

AUTOMATED FAULT TREE ANALYSIS OF A FAST BREEDER PRIMARY LOOP

by

MANOJ KUMAR

NETP
1987
M
KUM
AUT

TH
NETP/1987/M
K 96a



NUCLEAR ENGINEERING AND TECHNOLOGY PROGRAMME

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JANUARY, 1987

AUTOMATED FAULT TREE ANALYSIS OF A FAST BREEDER PRIMARY LOOP

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY



by
MANOJ KUMAR

to the

NUCLEAR ENGINEERING AND TECHNOLOGY PROGRAMME
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
JANUARY, 1987

- 2 Dec 1987
CENTRAL LIBRAR.
IT K₃-1
a.c. No A 98977

TH
005.3
K 9624

87-M-KUM-AUT

21/10/87
SRI

ii)

CERTIFICATE

This is to certify that the work presented in this thesis entitled, "AUTOMATED FAULT TREE ANALYSIS OF FAST BREEDER PRIMARY LOOP", has been carried out under our supervision and has not been submitted elsewhere for the award of degree.

K-Sri Ram
(K. Sri Ram)

Professor

Nuclear Engineering & Technology
Indian Institute of Technology
Kanpur-208016

Ratnakar Munshi
(P. Munshi)

Lecturer

December, 1986.

ACKNOWLEDGEMENTS

I am highly grateful to my guides Prof K Sri Ram and Mr. P. Munshi for their timely help and encouragement. But for their active participation and cooperation, this thesis work would not have been a success. It was only their affection and understanding that instilled a keen interest in my inner self and as a result I could work to the best of my abilities given the other constraints. They command high respect and regard in my heart.

My thanks are also due to my other teachers Prof. Sengupta and Dr. Kalra for maintaining high academic standards and helping me and my class mates as and when necessary.

I have high praise for other department staffs - Mr. Pathak, Mr. Tomar, Mr. Yadav and Mr. Gopal. They were highly cooperative and helpful.

I can not but praise my class-mates for their selfless attitude and active cooperation. It was only due to them that we could enjoy throughout the M.Tech. Program.

- MANOJ KUMAR

CONTENTS

	ABSTRACT	vi)	
CHAPTER	1	INTRODUCTION	1
CHAPTER	2	SYSTEM DESCRIPTION AND MODELLING	
	2.1	SNR 300 Details	5
	2.2	Fault Tree Construction	8
	2.3	Dats to be Used	10
	2.4	Reliability Program Objective	11
	2.5	Fault Tree	
CHAPTER	3	PROGRAM PLOMCP FOR FINDING MINIMAL CUTSETS	
	3.1	Definitions	13
	3.2	Redundancies to be Removed	13
	3.3	Algorithm	14
	3.4	Program Details	15
	3.5	Salient Features of the Program	17
	3.6	Limitations of the Program	18
	3.7	Operations to be Carried Out	19
CHAPTER	4	SYSTEM QUANTIFICATION - A THEORETICAL OVERVIEW	
	4.1	Some Definitions	20
	4.2	Fundamental Relations	23
	4.3	Relations among the Whole Process	24
	4.4	Laplace Transform Analysis	25
	4.5	System Analysis	26
	4.6	Alternative Formulation of System Analysis	33
CHAPTER	5	PROGRAM FOR SYSTEM QUANTIFICATION : MODKITT	
	5.1	Formulation for M out of N Identical System	35
	5.2	Flow Chart for System Analysis	36
	5.3	A Note on M out of N System	37
	5.4	Input to the Program	40
	5.5	Output	40
	5.6	Merits of the Program	41
	5.7	Demerits and Suggestion for Further Improvement	42
	5.8	Discussion of Result	43

	<u>Page</u>	
CHAPTER 6	ILLUSTRATIVE PROGRAM FOR MINIMAL PATH AND CUTSET FOR AN ELECTRICAL SYSTEM	
6.1	Introduction	46
6.2	Definition	46
6.3	Program Details	47
CHAPTER 7	FAULT TREE MODIFICATION PROGRAM	
7.1	Object	48
7.2	Difficulty	48
7.3	Approach	48
7.4	Program Details	49
7.5	Output	51
APPENDIX I	DATA TO BE USED	54
APPENDIX II	ALGORITHM FOR NUMERICAL INTEGRATION	58
APPENDIX III	ITERATION TECHNIQUE	59
REFERENCES		60

ABSTRACT

We ventured to develop a complete software package for fault tree analysis of any system be it mechanical, electrical or nuclear. In doing so attempt was made to improve upon the existing computer codes in terms of ease to use, CPU time, efficiency and software facilities. A particular attention was paid to enlarge the scope of the software. For example, almost all kinds of gates have been allowed using fault tree modification program. In fault tree quantification, 'M' out of 'N' type of system has been allowed and a new formulation has been done using Fussell's approximation. Save fault tree construction, all other processes in the fault tree analysis has been automated including the electrical systems associated in nuclear systems. We got unavailability at the end of 40 years to 0.5×10^{-7} for SNR-300 primary loop which is satisfactory according to 10 CFR 100 criterions. We also established that increase of redundancies improves the system but some components being more critical than others. Our computer codes can, hence, be of immense help in system design.

CHAPTER 1

INTRODUCTION

The present work is devoted to generate a complete software package for reliability analysis of any system using fault tree method in a most general form. Also discussed is the application of the package to primary coolant loop with emergency core cooling of a LMFBR for which model chosen is that of SNR-300 (A German 300 MW Fast Breeder Reactor). This package can be compared favourably with earlier similar packages in terms of its facilities, methodology, efficiency and CPU time on a computer.

The first program finds minimal cutsets for a fault tree containing upto 100 gates or 100 basic events. This number could be increased easily by increasing dimension of the concerned variables. MOCUS, on the other hand, can be used to find minimal cuts for upto 20 gates in a given tree. In this new program PCOMCP all basic events are coded by prime numbers and all logical operations and minimization processes are carried out by simple arithmetic operations. This reduces the storage requirement greatly since no character reading or logical operations are needed. A new method of providing input has been used in which number of lines or cards is no more than the number of basic events; thus simplifying the input process a great deal. In most of other programs

input is given gate by gate. Thus, number of input cards equals number of gates and hence a lengthy process. As in MOCUS, the maximum order of cutsets can also be specified in the input data.

Second program quantifies the fault tree, i.e. finds top event unavailability (probability) and other related parameters. This program is based on KITT [3] formulation but there is an improvement. The present program takes care of M out of N systems for basic events and top event. It is much easier to specify the values of M & N rather than modifying fault tree itself to take into account such systems. This kind of systems are mostly encountered due to large redundancy in nuclear systems to improve reliability. As in KITT1 and KITT2 bracketing procedure has been used to find maximum and minimum top event unavailability. In present program, at some places FUSSEL's approximation has been used when such parameter are not accurately desirable and formulation for exact value is otherwise difficult. Data could be provided both in terms of Lambda and Meu or Q (unavailability) and w (unconditional failure intensity). In latter case by bisection method of iteration, Lambda and Meu are obtained. There are three options for providing accuracy desired. Option 0 \Rightarrow take all cutsets into account. Option 1 \Rightarrow take maximum order of cutsets (specified). Option > 1 \Rightarrow specify the % accuracy desired. Integration uses Simpson's rule in

which decisional place accuracy is specified. It is obvious that program is most versatile and flexible in use.

Failure data and typical repair data for basic components have been obtained from WASH - 1400 report [1,2]
Nuclear Engg. & Design (1984 V 81-82, NED 1984 V 83),
[9]
Reliability technology by Green and Bourne and other

sources. From the analysis an unavailability of 0.5×10^{-7} has been obtained at the end of 40 year and it is seen that with time it decreases but not at a large rate for the primary coolant loop of LMFBR. This low unavailability has been obtained in LMFBR by using sufficient redundancies and standby systems.

Third program finds minimal cutsets for an electrical systems which contain large number of feed-backs, interconnection, and are represented by a ckt. graph rather than a fault tree. Entirely different approach is needed in this case. Electrical systems are integral part of nuclear power plant hence the interest. The program finds basic minimal path and the combinations which break these paths constitute the minimal cutsets [Chapter 6].

Last program is a fault tree modification program. It modifies the fault tree to take care of XOR, NOT, NAND, NOR gates. It utilises the gate equivalence for XOR and De Morgan's law for others. In some cases, level of the tree increases which makes the program slightly complex.

This kind of facility is not available in MOCUS or PREP-KITT. 'M' out of 'N' kinds of gates are taken care of at quantification level. Output of this program is a fault tree containing only AND and OR gates. Some of the basic events occur in negated form. It first determines the sub-branches which necessitates increase in the level of tree. All the basic event entries are modified accordingly. GATE equivalence and De Morgan's law suggest the rest of the procedure.

CHAPTER 2

SYSTEM DESCRIPTION AND MODELLING

The system we chose in this study for reliability analysis was primary sodium loop and emergency core cooling system of SNR-300 which is also called Kalkar nuclear power station (a joint German, Belgium and Dutch project). This design is very similar for many fast breeders and as we will see the main features are also valid for Clinch-River Breeder reactor plant and other fast breeder reactors.

2.1 SNR 300 Details (Only Primary & Intermediate Loop) [7]

The three parallel loops of the heat transfer system (HTS) transfer the heat produced in the reactor via intermediate heat exchangers and steam generators to the electricity producing system. They are furthermore used for the decay heat removal with the pump running at 5% of their nominal speed. Even with two loops out of service, natural convection in the third loop can remove the decay heat without reaching boiling at the core outlet. Even in cases leading to a failure of all main loops, the decay heat removal can take place through six emergency heat exchangers operating in parallel and located inside the reactor tank.

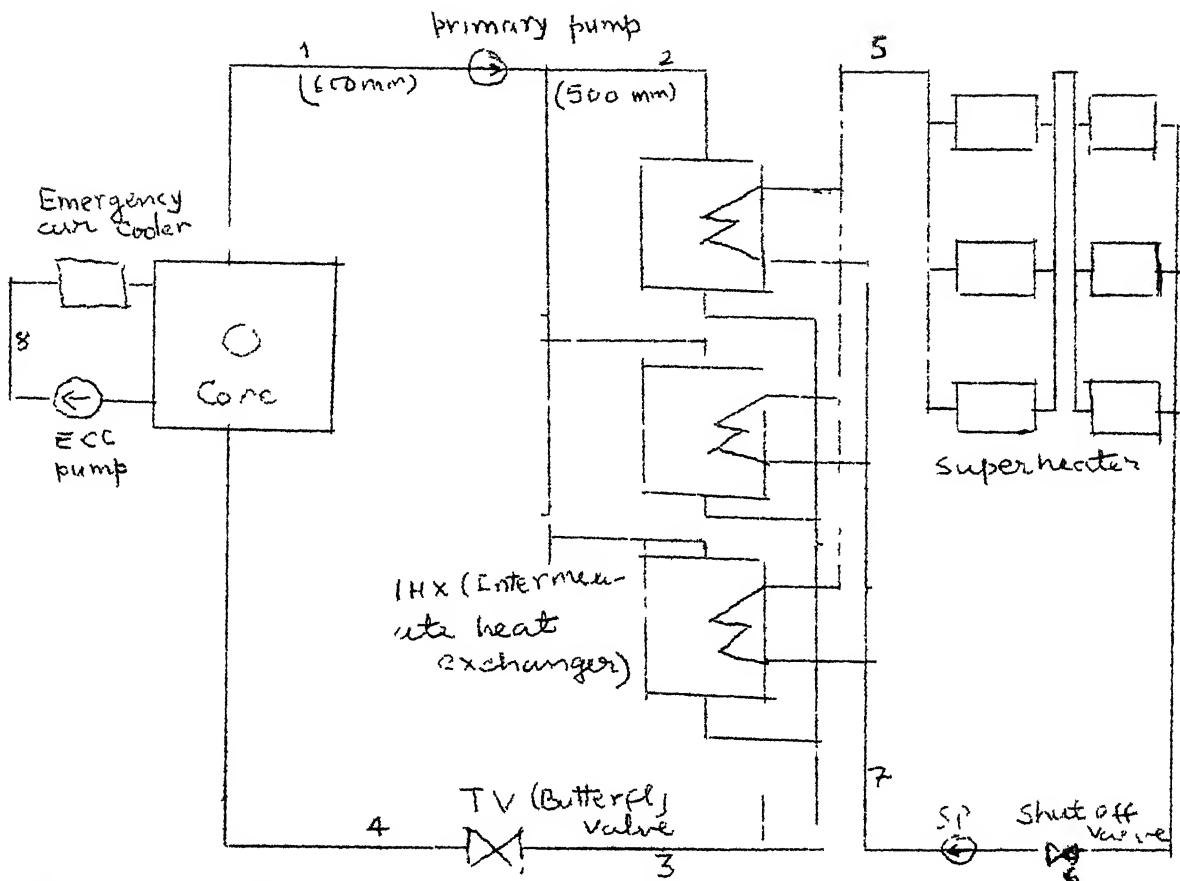


Figure : SNR-300 Prim. & Intermediate Coolant Loop

The pipes and components of heat transfer systems with the exception of the steam generator are using unstabilized austenitic stainless steel X8 Cr Ni 1811 (WN1.4948).

Each primary loop contains one circulating pump, a bank of three intermediate heat exchangers, one sodium flow meter and one throttle valve to reduce the coolant flow following a reactor shut down.

The coolant circulating pumps are positioned in the hot leg to provide a sufficient suction head at a low excess pressure prevailing in the upper part of the reactor

tank. The pumps are of variable speed, single stage, single-flow, radial type, arranged vertically with free sodium surface and argon cover gas atmosphere.

Heat exchangers are of straight tube design with a floating head at the lower end with the primary sodium flowing on the shell side, an arrangement that allows the tube bundle to be withdrawn without cutting the primary loop pipes.

The pipe connecting the reactor vessel to the pump has a diameter 600 mm which reduces to 500 mm for all other main pipes.

A butterfly valve located between IHX and the reactor vessel limits the sodium flow rate during decay heat removal operation, thus reducing thermal shocks.

All pipes except those in the annular region surrounding the reactor vessel are provided with an electric trace-heating system. The pipes in the reactor vessel, the pumps and IHX's are preheated by hot nitrogen.

The pipes of the three primary loops are laid out symmetrically in the reactor cell, from which they run to the three parallel primary cells. This arrangement cases shielding of the piping penetrations and reduces the activation of the structural material and of the secondary sodium.

2.2 Fault Tree Construction

First of all referring to SNR-300 figure we divide loop into two parts.

(a) Primary Loop - Consisting of pipe segments 1,2,3,4, primary pump, Butterfly valve and heat exchanger.

(b) Secondary Loop - Consisting of pipe segments 5,6,7, shut off valve, superheater, evaporator and secondary pump.

Then following observations were made.

(i) Pump has two kinds of failures - (a) single ended or minor failure; (b) double ended or major failure.

(ii) Different pipe segments can in general (and in fact they are so) be of different diameters hence their failure data is expected to be different from one another. As a result, these pipe segments break or leak must be considered as separate basic events.

(iii) Heat exchangers, Butterfly valves, primary pump, secondary pump, shut off valve, superheater, evaporator have redundant duplications intended to improve reliability.

(iv) Only when primary or intermediate heat transport system and emergency core cooling system fail, there is uncontrolled rise in core temperature.

(v) Since major components have redundancies, there must be maintenance when one or more of redundant and main component fail. Hence repair rate must be taken.

(vi) Primary pump is a sodium pump and is more sophisticated. Its data must be chosen carefully.

(vii) There are three identical primary loops hence over all system is 1 out of 3.

Based on above information we make further subdivision of the system as follows: Primary loop is divided into two parts, (a) Valve line consisting of pipe segments 3,4 and Butterfly valve, and (b) Pump line consisting of pipe segments 1,2, primary pump and intermediate heat exchangers. Any of these parts' failure will lead to primary loop failure. Valve line failure occurs when either there is no supply from Butterfly valve or pipe length 4 has failed. Butterfly valve supply is cut off when either pipe length 3 is cut off or Butterfly valve is stuck. Pump line failure occurs when either there is no input to heat exchanger or heat exchanger fails. There is no input to heat exchanger if either pipe segment 2 fails or no supply from primary pump. There is no supply from primary pump if pipe segment 1 fails or pump fails.

Intermediate loop fails if either valve line (consisting of pipe length 6 and 7, shut off valve and

sec. pump) fails or super heater, evaporator line fails (consisting of superheater, evaporator or pipe length 5). Valve line fails if either pipe length 6 fails or subsystem from valve to heat exchanger fails. The later fails if pipe length 7 fails or pump and valve system fails. Pump and valve system fails if either of these fails. Superheater, evaporator line fails if pipe length 5 fails or any of superheater and evaporator fails.

Emergency core cooling system fails if either air cooler fails or there is no supply to air cooler. There is no supply to air cooler if either emergency pump fails or pipe length fails.

2.3 Data to be Used

Based on Appendix I, we decided to use the following data:

Component	$\lambda \text{ hr}^{-1}$	$\mu \text{ hr}^{-1}$
Emer. cooling pipe (single ended failure)	2.0×10^{-9}	1.2×10^{-7}
Emer. cooling pipe (double ended failure)	2.0×10^{-10}	1.0×10^{-8}
Other pipes (single ended)	1.0×10^{-10}	5.0×10^{-9}
Other pipes (double ended)	1.0×10^{-12}	4.0×10^{-11}
Valve (Butterfly)	1.6×10^{-5}	1.0×10^{-4}
Primary pump	9.17×10^{-7}	1.0×10^{-5}
Heat exchanger	4.184×10^{-6}	2.0×10^{-5}

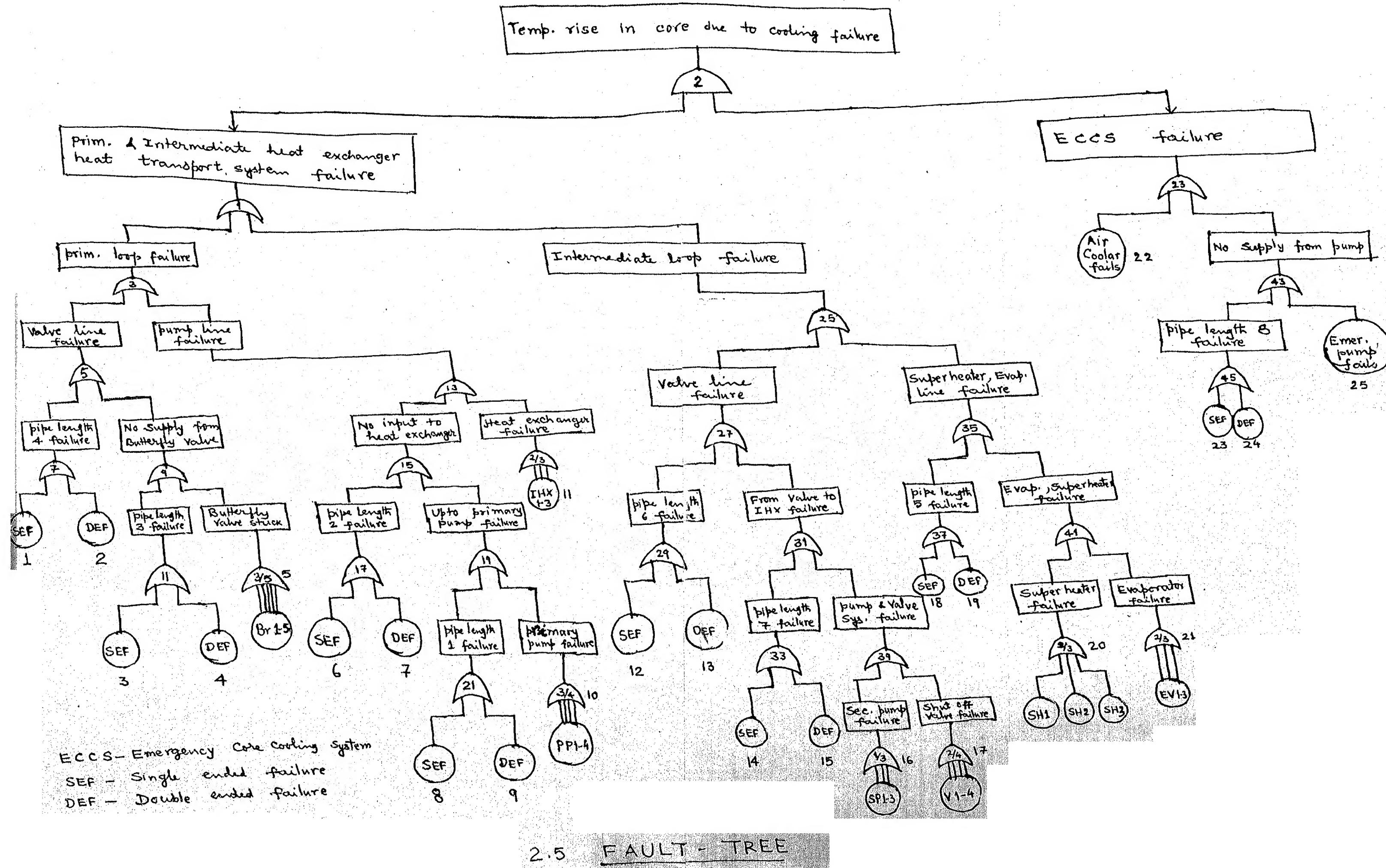
Component	$\lambda \text{ hr}^{-1}$	$\mu \text{ hr}^{-1}$
Secondary pump	2.5×10^{-6}	2.5×10^{-5}
Shut off valve	3.0×10^{-5}	3.0×10^{-4}
Super-heater	8.0×10^{-6}	5.0×10^{-5}
Evaporator	8.0×10^{-6}	5.0×10^{-5}
ECC pump	2.5×10^{-6}	2.5×10^{-4}
Air cooler	3.0×10^{-6}	5.0×10^{-4}

2.4 Reliability Program Objectives

We aim at ensuring that the likelihood of exceeding the nuclear radiation dose guidelines of 10CFR 100 [2] at the plant site boundary should not be greater than 10^{-6} per reactor operating year that in-place core-coolable geometry will be lost [8]. Loss of in-place coolable core geometry is a failure criterion which is used to characterize very low probability events. If core coolable geometry is lost, there is no assurance that significant core damage could not occur, even though there is still a low probability that site boundary dose guidelines could be exceeded since the plant includes a number of containing barriers, one of which is the containment structure. The probability of occurrence of potential initiators of loss of coolable geometry can be controlled by design. It then becomes the task of the Reliability Program to assure the

high reliability of systems necessary to prevent the onset of such initiating events. Basic tasks of the Reliability Program are (1) to identify those extremely unlikely events having the potential to produce loss of coolable core geometry (2) to ensure through reliability engineering design that all such events are of sufficiently low probability to meet the goal set (3) to confirm this high reliability through analytic assessment and testing.

An initial reliability allocation against the overall goal was made for potential initiators which might threaten coolable geometry [8]. The division of the overall numeric goal among safety systems was based on the functional role of each system and its predicted relative failure potential. On the basis of this, the preliminary unreliability allocation established for shutdown heat removal system [8] was $< 8 \times 10^{-7}$ failures per year.



2.5 FAULT - TREE

CHAPTER 3

PROGRAM PCOMCP FOR FINDING MINIMAL CUTSETS

3.1 Definition

Cutsets - A cutset is a collection of basic events; if all these basic events occur, the top event is guaranteed to occur.

Pathset - It is a collection of basic events and if none of the events in the set occur, the top event is guaranteed to not occur.

Minimal cutset - A minimal cutset is such that if any basic event is removed from the set, the remaining events collectively are no longer a cutset. This is obtained from cutset by removing cuts which are redundant according to some rules laid down below.

Minimal Pathset - is a path set such that if any basic event is removed from the set, the remaining events collectively are no longer a path set.

3.2 Redundancies to be removed for Generation of Minimal Cutset

(1) Redundant Factor

If in any cut we have $A \cdot A$, A or A like term replace it by A .

(ii) Subset Redundancy

If two cuts are such that one is subset of other remove the superset, i.e. $A + AB = A$.

(iii) Term Redundancy

If two terms are repeated in the cutset then remove the repeated term, $AB + AB = AB$.

3.3 Algorithm

The program is based on the unique factorization properties of natural numbers stated in the form of following theorem.

Unique Factorization Theorem [4]

Every natural number greater than 1 can be expressed as a product of prime factors in one and only ^{one} way, apart from the order in which the factors are written.

Our strategy is to assign one prime number to each basic event, doing arithmetic operations for logical GATES, removing redundancies in cutsets and finally decode them back.

A particular combination of basic events can be expressed uniquely as a single number which is equal to the product of prime numbers corresponding to the basic events.

3.4 Program Details

We follow the following steps:

- (i) Assignment of one prime number to each basic event.
- (ii) Go bottom up.
- (iii) Go up level by level (to be described). A level is the number of gates from top after which the event is encountered.
- (iv) For each of OR gate encountered we preserve all the inputs to gate in an array.
- (v) For each AND gate we multiply the input codes.
- (vi) Of the final array, if two number is multiple of or equal other larger/~~is~~ removed. This is to remove of redundancy No. (ii) and (iii).
- (vii) Of the final set, each numbers is prime factorized. If any factor is repeated in the factor set they are removed. The factor set when decoded gives minimal cutset. This step removes the redundancy (i).
- (viii) If number of prime factors increase the maximum order of cutsets wanted, that cut is also removed.
- (ix) AND gate is designated by an even number to the gate and OR gate by an odd number.

Method of inputting the tree structure is by assigning a series of integers which represent the gates linking the top event to a primary event. The length of each series is equal to maximum level (i.e. maximum number

of gates from Top event to basic event) plus one. The gate number 0 indicates that there is no gates at that position. The order of row corresponds to the configuration of a fault tree.

Level by level is done as follows.

First line of input data is read and level is decided. The number is stored in the matrix element $N(L,1)$ where L is level number. Second line of input data is then read. If level number is same as previous number, the last gate operation is done. If last gate is OR gate (i.e. gate number is odd), it is stored in $N(L,2)$ i.e. in the $N(L,x)$ which is vacant (i.e. zero) and x is minimum. On the other hand if gate is AND gate (i.e. gate number is even) all $N(L,x)$ elements are multiplied by the current line basic event number. Once the gate operation is done level is reduced by one and this number is stored. Now the third line is read. If the level number equals the level number stored, the gate operation is done according to rules aforesaid and gate type indicated by the gate number being odd or even at the level's place from the Top gate number row wise. Level number is reduced by one and this level is stored. In cases when the current level is not equal to the stored level number, the current basic event coded number is stored in $N(L',1)$ and next line read, if level is again not equal to previous level number, this basic even number is stored in $N(L'',1)$.

and this is carried till the two successive levels are equal. Gate operation is done and level number is again reduced. If this level pointer is equal to previous level number, proper operation done and this process is repeated till all the numbers are in $N(1,x)$ where x varies from 1 to No. of OR gates in the fault tree.

To remove the redundancies (ii) and (iii), the final set of numbers $N(1,x)$ $\forall x <$ No. of OR gates, is checked for multiplicity. If any number in this set is multiple of any other, the larger (or equal) is set equal to zero. Once this procedure is carried out supersets are removed. Next for each non-zero elements in $N(1,x)$ set, prime factors are found. If any prime factor is repeated in any element in $N(1,x)$, this repeated factor is removed, i.e. one of repeated factors are kept. This process ensures the removal of type (i) redundancy. Corresponding to each number in set $N(1,x)$, the set of non-repeated prime factors when decoded give minimal cutset elements. The number of minimal cutsets equal the number of nonzero element in set $N(1,x)$.

3.5 Salient Features of the Program

(i) Coding by prime numbers ease out the minimization of cutset procedure.

(ii) Logical operations carried out by arithmetic and matrix operation.

(iii) Actual basic event No. when coded by prime numbers does not increase the CBE value very much as compared to actual basic event No. (CBE=Coded Basic Event).

(iv) The maximum order of cutsets could be specified.

(v) Level by level operation is actual practice reduces the total number of operation required.

(vi) Because of (i), (iii) and (v) the program is efficient and takes less CPU time.

(vii) Memory storage required is less than many other minimal cutset enumeration programs.

(viii) The maximum number of OR gates and input right now could be 100. This can be increased by simply increasing the dimension of some arrays in the program.

3.6 Limitations of the Program

(i) All gates must have only two inputs. Only the top gate can have 3 or more inputs.

(ii) As it is the program takes care of only AND and OR gates. Some more gates can be taken into account as mentioned below:

(a) NOT gate - the basic event itself is changed. It is considered independent event with $R = 1$ - not inverted event reliability.

(b) INHIBIT Gate - It is considered an AND gate. With conditional event as a basic event.

(c) Priority AND - It is basically AND gate with some sequencing; so it is treated as AND gate.

3.7 Operations to be Carried Out

The operation can be summarized as below:

- (a) A line is read and level is decided.
- (b) If level is equal to previous step level no.

Logical operation is to be done otherwise CBE is stored in $N(L, 1)$.

(c) Logical operation is carried out as follows: If it is an OR gate the CBE is to be stored in vacant place of $N(L, x)$. On the other hand, if it is an AND gate all the elements of $N(L, x)$ are multiplied by CBE,

(d) If any logical operation is done LP is decreased by 1. If NG (LP) is even, (i.e. AND gate) all combination of multiplication of $N(LP, x)$ and $N(LP + 1, x)$ is carried out and stored in $N(LP, x)$. In cases when NG (LP) is odd, All elements of $N(LP + 1, x)$ stored in vacant (i.e. zero) places of $N(LP, x)$. All elements of $N(LP + 1, x)$ reduced to zero.

(e) LP decreased further and (d) line operation carried out till LP is zero.

(f) When all of input data is over, factorization, minimization and decoding is done as outlined above. If data is not over go to (a).

CHAPTER 4

SYSTEM QUANTIFICATION - A THEORETICAL OVERVIEW

4.1 Some Definitions [13] and Symbol Meanings

(a) Reliability at time $t = R(t)$.

The probability that the component experiences no failure during the time interval $(0, t]$ given that the component was repaired at time zero.

(b) Unreliability at time $t = F(t) = 1 - R(t)$ (4.1)

(c) Failure density at time $t = f(t)$

The first order derivative of $F(t)$.

(d) Failure rate = $r(t)$.

The probability that the component experiences a failure per unit time at time t given that the component was repaired at time zero and has survived to time t .

(e) Mean time to failure = MTTF

The expected value of time to failure, i.e. mean of the span of time from repair to first failure

$$\text{MTTF} = \int_0^{\infty} t f(t) dt \quad (4.2)$$

(f) Repair probability at time $t = G(t)$.

The probability that the repair is completed before time t given that the component failed at time zero.

(g) Repair density of $G(t) = g(t)$

The first order derivative of $G(t)$.

(h) Mean time to repair = MTTR.

The expected value of time to repair, i.e. mean of length of time from the failure to the succeeding first repair

$$\text{MTTR} = \int_0^{\infty} t g(t) dt \quad (4.3)$$

(i) Repair rate = $m(t)$.

The probability that the component is repaired per unit time at time t given that the component failed at time zero and has been failed to time t .

(j) Availability at time t = $A(t)$.

The probability that the component is normal at time t given that it was as good as new at time zero.

$$A(t) \geq R(t).$$

(k) Unavailability at time t = $Q(t)$

The probability that a component is in the failed state at time t , given that it jumped into the normal state at time zero

$$Q(t) \leq F(t)$$

(l) Conditional failure intensity = $\lambda(t)$

The probability that the component fails per unit time at time t , given that it is in the normal state at time zero and is normal at time t .

(m) Unconditional failure intensity at time t = $w(t)$

The probability that a component fails per unit time at

t , given that it jumped into the normal state at time zero.

(n) Expected number of failures (ENF) = $W(t, t+dt)$

Expected number of failures during $[t, t+dt]$, given that the component jumped into the normal state at time zero.

(o) $W(t_1, t_2) = \text{ENF}$ over a period.

Expected number of failures during (t_1, t_2) , given that the component jumped into the normal state at time zero.

$$W(t_1, t_2) = \int_{t_1}^{t_2} w(t) dt \quad (4.4)$$

(p) Conditional repair intensity = $\mu(t)$

The probability that a component is repaired per unit time at time t , given that it jumped into the normal state time zero and is failed at time t .

(q) Unconditional repair intensity at time $t = v(t)$

The probability that the component is repaired per unit time at time t , given that it jumped into the normal state at time zero.

(r) Expected number of repairs in an interval

$$= V(t_1, t_2)$$

Expected number of repairs during $[t_1, t_2]$, given that the component jumped into the normal state at time zero.

$$V(t_1, t_2) = \int_{t_1}^{t_2} v(t) dt \quad (7.5)$$

4.2 Fundamental Relations [14]

$$A(t) = R(t) \text{ for non-repairable components}$$

$$\lambda(t) = r(t) \text{ for non-repairable components}$$

$$R(t) = \int_t^{\infty} F(u) du \quad (4.6)$$

$$r(t) = \frac{f(t)}{1-F(t)} \quad (4.7)$$

$$R(t) = \exp \left[- \int_0^t r(u) du \right] \quad (4.8)$$

$$f(t) = r(t) \exp \left[- \int_0^t r(u) du \right] = r(t)R(t) \quad (4.9)$$

$$m(t) = \frac{g(t)}{1-G(t)} \quad (4.10)$$

$$G(t) = 1 - \exp \left[- \int_0^t m(u) du \right] \quad (4.11)$$

$$g(t) = m(t)[1-G(t)] = m(t) \exp \left[- \int_0^t m(u) du \right] \quad (4.12)$$

If constant failure rate $= \lambda = r(t)$ and non-repairable

$$\lambda(t) = -\frac{1}{R(t)} \frac{dR(t)}{dt} \quad (4.13)$$

$$R(t) = e^{-\lambda t} \quad (4.14)$$

$$F(t) = 1 - e^{-\lambda t} \quad (4.15)$$

$$f(t) = \lambda e^{-\lambda t} \quad (4.16)$$

$$MTTF = \frac{1}{\lambda} \quad (4.17)$$

If constant repair rate $= \mu = m(t)$

$$G(t) = 1 - e^{-\mu t} \quad (4.18)$$

$$g(t) = \mu e^{-\mu t} \quad (4.19)$$

$$MTTR = \frac{1}{\mu} . \quad (4.20)$$

4.3 Relations among the Whole Process

(a) Unconditional intensities $w(t)$ and $v(t)$. The component which fail during $[t, t+dt]$ are of two types:

(i) Type 1: A Component which was repaired during $[u, u+du]$, has been normal at time t , and fails during $[t, t+dt]$, given that the component jumped into the normal state at time zero.

Probability for such component is $v(t) du \cdot f(t-u)dt$ since $v(u)du$ = probability that the component is repaired during $[u, u+du]$, given that it is as good as new at time zero and $f(t-u)dt$ = the probability that the component has been normal to time t and failed during $(t, t+dt)$, given that it was as good as new at time zero and was repaired at time u .

(ii) Type 2: A component which has been normal to time t and fails during $[t, t+dt]$, given that it jumped into the normal state at time zero.

Probability of second type of components is $f(t)dt$
 $w(t) dt$ = Probability that the component fails during $[t, t+dt]$, given that it jumped into the normal state at time zero
 $w(t)dt = f(t) dt + \int_t^t dt \int_0^t f(t-u) v(u) du$
or $w(t) = f(t) + \int_0^t f(t-u) v(u) du$ (4.21)

(b) Second Relationship: The component which are repaired during $[t, t+dt]$ is one which was failed during $[u, u+du]$, has been failed to time t and repaired during

$[t, t+dt]$, given that the component jumped into the normal state at time zero.

The probability of such components is $w(t) \int_0^t g(t-u) du dt$

$$v(t) dt = dt \int_0^t g(t-u) w(u) du$$

$$\therefore v(t) = \int_0^t g(t-u) w(u) du \quad (4.22)$$

(c) Unavailability:

$$Q(t) = W(0, t) - V(0, t) \quad (4.23)$$

= Number of failures - Number of repairs
at time t

(d) Failure intensity $\lambda(t)$.

$$\lambda(t) = \frac{w(t)}{1-Q(t)} \quad (4.24)$$

(e) Repair intensity $\mu(t)$

$$\mu(t) = \frac{v(t)}{Q(t)} \quad (4.25)$$

4.4 Laplace Transform Analysis for the Whole Process

Laplace Transform of (4.16)

$$\begin{aligned} L[f(t)] &= \frac{\lambda}{s+\lambda} \\ L[g(t)] &= \frac{\mu}{s+\mu} \quad (4.19) \end{aligned}$$

Laplace transform of (4.21) and (4.22)

$$\begin{aligned} L[w(t)] &= L[f(t)] + L[f(t)] \cdot L[v(t)] \\ L[v(t)] &= L[g(t)] \cdot L[w(t)] \end{aligned}$$

Using the $L[f(t)]$ and $L[g(t)]$

$$L[w(t)] = \frac{\lambda}{s+\lambda} + \frac{\lambda}{s+\lambda} L[v(t)]$$

$$L[v(t)] = \frac{\mu}{s+\mu} L[w(t)]$$

Solving the two equations for $L w(t)$ and $L v(t)$

$$L[w(t)] = \frac{\lambda\mu}{\lambda+\mu} \left(\frac{1}{s}\right) + \frac{\lambda^2}{\lambda+\mu} \left(\frac{1}{s+\lambda+\mu}\right)$$

$$L[v(t)] = \frac{\lambda\mu}{\lambda+\mu} \left(\frac{1}{s}\right) - \frac{\lambda\mu}{\lambda+\mu} \left(\frac{1}{s+\lambda+\mu}\right)$$

Taking Laplace inverse

$$w(t) = \frac{\lambda\mu}{\lambda+\mu} + \frac{\lambda^2}{\lambda+\mu} e^{-(\lambda+\mu)t} \quad (4.26)$$

$$v(t) = \frac{\lambda\mu}{\lambda+\mu} - \frac{\lambda\mu}{\lambda+\mu} e^{-(\lambda+\mu)t} \quad (4.27)$$

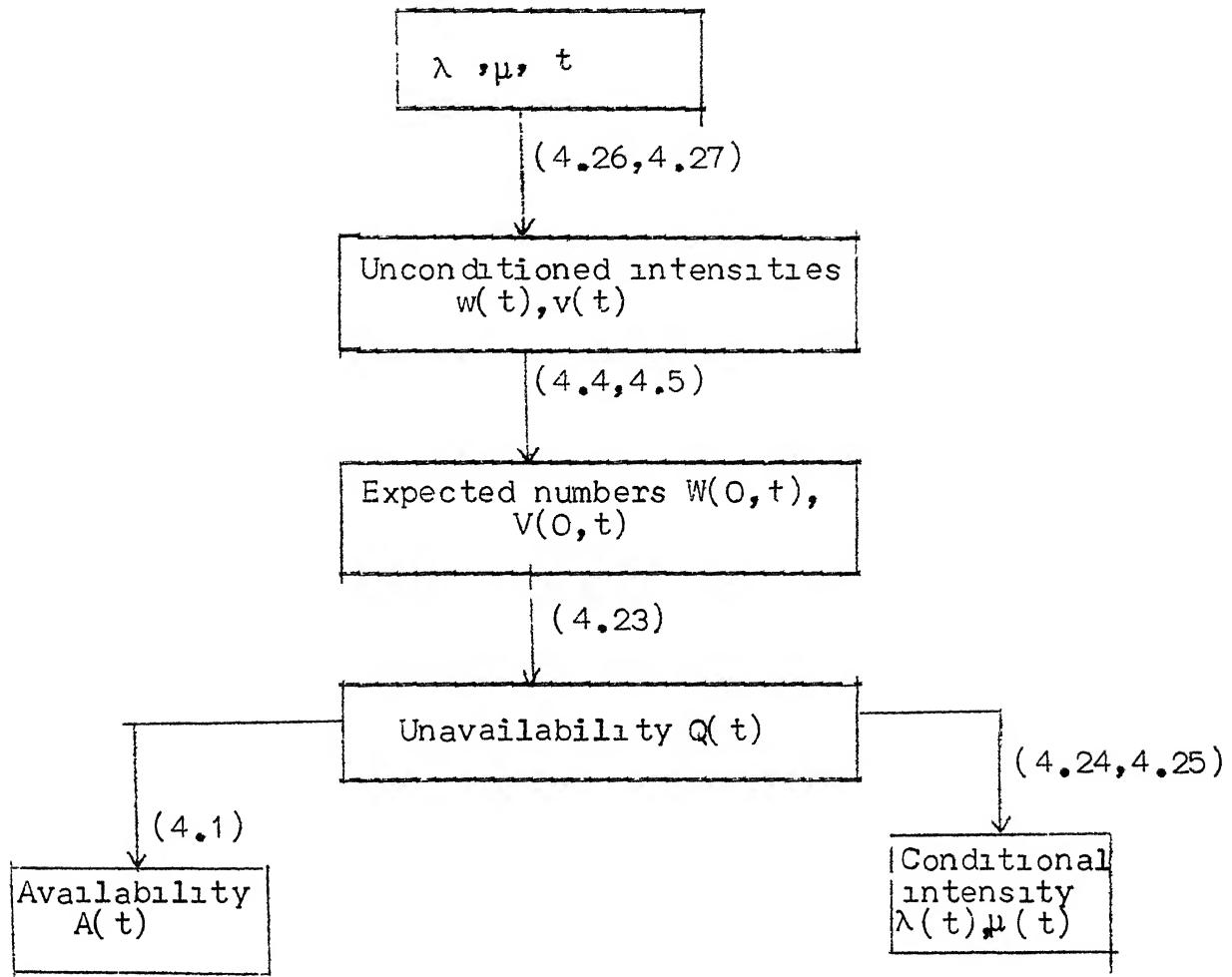
$$W(0, t) = \frac{\lambda\mu}{\lambda+\mu} t + \frac{\lambda^2}{(\lambda+\mu)^2} (1 - e^{-(\lambda+\mu)t}), \quad (4.28)$$

$$V(0, t) = \frac{\lambda\mu}{\lambda+\mu} - \frac{\lambda\mu}{(\lambda+\mu)^2} (1 - e^{-(\lambda+\mu)t}), \quad (4.29)$$

$$Q(t) = W(0, t) - V(0, t) = \frac{\lambda}{\lambda+\mu} (1 - e^{-(\lambda+\mu)t}) \quad (4.30)$$

4.5 System Analysis

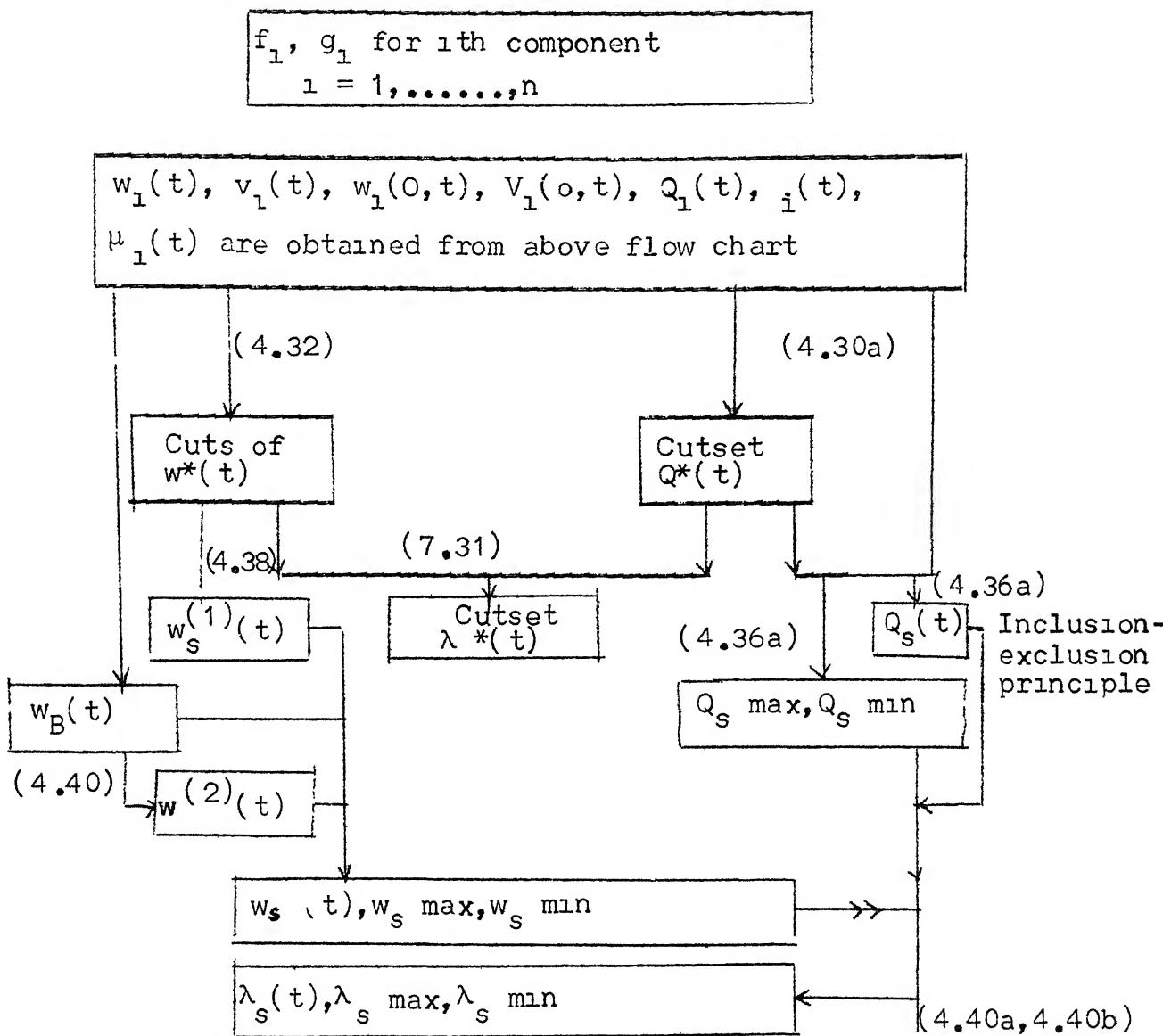
A. Component analysis is done on the basis of following flow chart:



Modified version of above flow chart will be discussed in the next chapter which we used in our program.

KITT - The code is an application of kinetic Tree theory and handles independent basic events which are non-repairable or repairable, provided they have constant failure rates and constant repair rates

B. Flow Chart for KITT Computations



Exact time-dependent reliability parameters are determined for each basic events and cutsets, but for the system as a whole the parameters are obtained by upper or lower bound approximations or by bracketing. In bracketing procedure, the various upper and lower bounds can be obtained as close to each other as desired, and thus the exact value

for system parameters are obtained if the user so chooses.

$w(t)$, $v(t)$ are obtained by equations (4.21) and (4.22). For this purpose, numerical integration is used. $w(0,t)$, $v(0,t)$ are obtained by another integration for given t . Q is found by equation (4.23).

Minimal Cutset Parameters:

(a) Unavailability - A cutset is occurring if all the basic events in the cutset are occurring

$$Q^*(t) = \prod_{j=1}^n Q_j(t) \quad (4.30a)$$

(b) Conditions failure intensity and Unconditional failure intensity:

$$\lambda^*(t) = \frac{w^*(t)}{1 - Q^*(t)} \quad (4.31)$$

where

$$w^*(t) = \sum_{j=1}^n w_j(t) \prod_{\substack{l=1 \\ l \neq j}}^n Q_l(t) \quad (4.32)$$

(c) Similarly

$$v^*(t) = \sum_{j=1}^n v_j(t) \prod_{\substack{l=1 \\ l \neq j}}^n [1 - Q_l(t)] \quad (4.33)$$

$$\mu^*(t) = \frac{v^*(t)}{Q^*(t)} \quad (4.34)$$

(d) Also

$$w^*(0,t) = \int_0^t w^*(u) du \quad (4.35)$$

$$v^*(0,t) = \int_0^t v^*(u) du \quad (4.36)$$

System Parameters.

(a) Unavailability

Let $d_i =$ event that all the basic events of i th minimal cutset exist at time t .

The i th minimal cutset failure exist at time t .

By inclusion-exclusion principle

$$\begin{aligned}
 Q_s(t) &= \Pr\left(\bigcup_{i=1}^{N_c} d_i\right); \quad N_c = \text{No. of minimal cutsets} \\
 &= \sum_{i=1}^{N_c} \Pr(d_i) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} \Pr(d_i \cap d_j) + \dots \\
 &\quad + \dots + (-1)^m \sum_{1 \leq i_1 < i_2 < \dots < i_m \leq N_c} \Pr(d_{i_1} \cap d_{i_2} \cap \dots \cap d_{i_m}) \\
 &\quad + \dots + (-1)^{N_c-1} \Pr(d_1 \cap d_2 \cap \dots \cap d_{N_c})
 \end{aligned}$$

The m th term is the contribution to $Q_s(t)$ from m minimal cut set failures existing simultaneously at time t

$$\begin{aligned}
 Q_s(t) &= \sum_{i=1}^{N_c} Q_i(t) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} \pi_{i,j} Q(t) + \dots \\
 &\quad + (-1)^{m-1} \sum_{1 \leq i_1 < i_2 < \dots < i_m \leq N_c} \pi_{i_1, \dots, i_m} Q(t) + \dots \\
 &\quad + (-1)^{N_c-1} \pi_{1, \dots, N_c} Q(t) \tag{4.36a}
 \end{aligned}$$

where π_{i_1, \dots, i_m} is the product of $Q(t)$'s for the basic events in cutset i_1 or i_2, \dots, i_m .

Bracketing: If in above expression, only first term is taken, it is upper bound for $Q_s(t)$. If first two terms taken, it is lower bound for $Q_s(t)$. If 1st three terms taken, it is better upper bound for $Q_s(t)$ and so on.

Upper bound alternative expression is due to Esary and Proschan = $1 - \sum_{i=1}^{N_c} [1 - Q_i * (t)]$.

This upper bound estimate is sometimes conservative estimate, but it is exact when the cutsets are disjoint sets of basic events.

(b) Unconditional failure intensity

$w_s(t)$ = Expected number of times the top event occurs at time t , per unit time

$w(t; 1, \dots, m)$ = The unconditional failure intensity for a mode of failure which has as its basic failures the basic failures which are common members to all the mode failures $1, \dots, m$

$$\begin{aligned} w_s^{(1)}(t) &= \sum_{i=1}^{N_c} w_i * (t) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} w(t; i, j) \pi_{i,j} Q(t) \\ &+ \sum_{i=3}^{N_c} \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} w(t; i, j, k) \pi_{i,j,k} Q(t) + \dots \\ &+ (-1)^{N_c-1} w(t; 1, \dots, N_c) \pi_{1, \dots, N_c} Q(t) \quad (4.38) \end{aligned}$$

A = The event that one or more of cutset failure occur at time t .

$$B = \bar{A}$$

$$w_B(t; 1, \dots, m) dt = \Pr(e_1 \cap \dots \cap e_m \cap B)$$

$$\begin{aligned} &= \sum_{i=1}^{N_c} \Pr(e_1 \cap \dots \cap e_m \cap d_i) - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} \Pr(e_1 \cap \dots \cap e_m \cap \\ &\quad \dots \cap d_i \cap d_j) \dots + (-1)^{N_c-1} \Pr(e_1 \cap \dots \cap e_m \cap d_1 \cap \dots \cap d_{N_c}) \end{aligned}$$

$$w_s^{(2)} = \sum_{i=1}^{N_c} w_B(t; i) dt - \sum_{i=2}^{N_c} \sum_{j=1}^{i-1} w_B(t; i, j) + \dots \\ + (-1)^{N_c-1} w_B(t; 1, \dots, N_c) \quad (4.40)$$

Bounds:

$$(1) \quad w_s(t)_{\min} = w_s^{(1)}(t)_{\min} - w_s^{(2)}(t)_{\max}$$

$$w_s(t)_{\max} = w_s^{(1)}(t)_{\max} - w_s^{(2)}(t)_{\min}$$

(ii) If N_c is even.

$$Q_s(t) = Q_s(t)_{\min}$$

$$w_s(t) = w_s(t)_{\min}$$

$$\lambda_s(t) = \lambda_s(t)_{\min}$$

If N_c is odd:

$$Q_s(t) = Q_s(t)_{\max}$$

$$w_s(t) = w_s(t)_{\max}$$

$$\lambda_s(t) = \lambda_s(t)_{\max}$$

$$(iii) \lambda_s(t)_{\max} = \frac{w_{s,\max}}{1 - Q_{s,\max}} \quad (4.40a)$$

$$\lambda_s(t)_{\min} = \frac{w_{s,\min}}{1 - Q_{s,\min}} \quad (4.40b)$$

(c) Integrated number of failures

$$W_s(0, t) = \int_0^t w_s(u) du \quad (4.41)$$

4.7 Alternative Formulation of System Analysis - Short-Cut Technique

J.B. Fussell formulated this technique which is basically back-of-the-envelope guesstimate. It requires as input failure and repair rates for basic events and minimal cutsets. It assumes exponential distribution of and independent of component failures.

(a) Component Level Analysis:

$$Q_1 = 1 - e^{-\lambda_1 t} = \lambda_1 t \quad (4.42)$$

if $\lambda_1 t < 0.1$

If components repairable

$$Q_1 = \frac{\lambda_i}{\lambda_1 + \mu_1} (1 - e^{-(\lambda_i + \mu_i)t})$$

as t becomes large and if $\frac{\lambda_1}{\mu_1} \ll 0.1$

$$Q_1 \approx \frac{\lambda_i}{\lambda_1 + \mu_1} \approx \frac{\lambda_1}{\mu_1} \text{ if } t > \frac{2}{\mu_1} \quad (4.43)$$

(b) Cutset level Analysis:

$$Q_1 * (t) = \prod_{i=1}^n Q_i \quad (4.44)$$

Using equations (4.32) and (4.24)

$$w_1 * (t) \approx \sum_{j=1}^n [1 - Q_j(t)] \lambda_j(t) \prod_{\substack{l=1 \\ l \neq j}}^n Q_l(t)$$

Substituting equation (4.44) here,

$$w_1 * (t) \approx Q_1 * (t) \sum_{j=1}^n \frac{\lambda_j}{Q_j}(t) \quad (4.45)$$

noting $1 - Q_j(t) \approx 1$

$$\lambda * (t) = \frac{w_1 * (t)}{1 - Q_1 * (t)} \quad (4.46)$$

(c) System Level Analysis using bounding Procedures:

$$Q_s(t) \approx \sum_{i=1}^{N_c} Q_i * (t) \quad (4.47)$$

$$\lambda_s \approx \sum_{i=1}^{N_c} \lambda_i * \quad (4.48)$$

$$w_s(t) \approx \sum_{i=1}^{N_c} w_i * (t) \quad (4.49)$$

$$\text{or } \lambda_s \approx \frac{w_s(t)}{1 - Q_s(t)} \cdot \quad (4.50)$$

CHAPTER 5

PROGRAM FOR SYSTEM QUANTIFICATION : MODKITT

5.1 Formulation for m Out of n Identical Systems

(a) Unavailability:

m out of n system: Top event unavailable if any of m or more systems out of n systems are unavailable.

By binomial distribution,

$$Q_s(t) = \sum_{k=m}^n \binom{n}{k} Q^k (1-Q)^{n-k} \quad (5.1)$$

(b) Unconditional and conditional intensities for m out of n systems. There are $\binom{n}{m}$, mth order cutsets; $\binom{n}{m+1}$, (m+1)th order cutsets and so on. If we neglect higher order cutsets we are left with $\binom{n}{m}$, mth order cutsets. For finding minimal cutsets, we must do so.

Using equation (4.32) for any cutset of mth order

$$w^*(t) = m Q^{m-1} w$$

Since there will be m identical terms with value $w Q^{m-1}$
Finally using Fussell's approximation [13] equation (4.49)

$$\begin{aligned} w_s(t) &= \binom{n}{m} m Q^{m-1} w \\ &= \frac{n!}{m! (n-m)!} m Q^{m-1} w \end{aligned} \quad (5.2)$$

$$w_s(t) = \frac{n!}{(m-1)! (n-m)!} Q^{m-1} w \quad (5.2)$$

$$\lambda_s(t) = \frac{w_s(t)}{1-Q_s(t)} \quad (5.3)$$

(c) Unconditional and conditional repair intensities using eqn.(4.33)

$v^*(t) = m v(t) \cdot (1-Q)^{m-1}$ for any of $\binom{n}{m}$ cutsets of m th order.

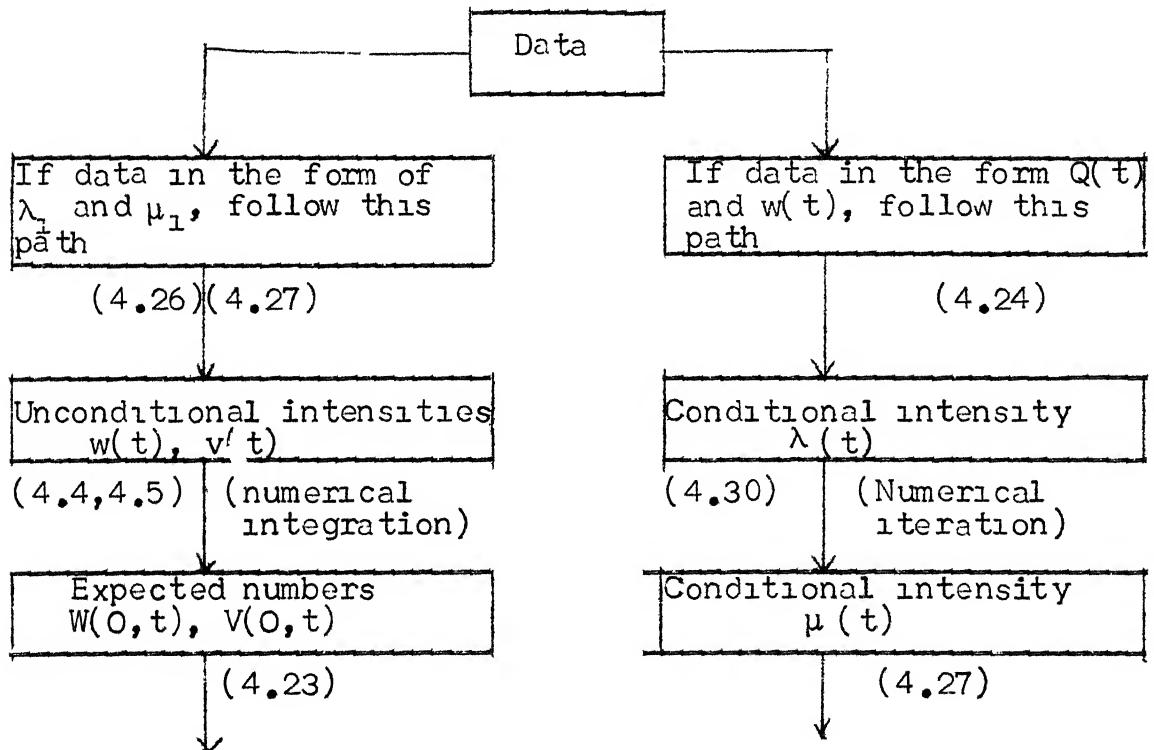
Using Fussell's approximation

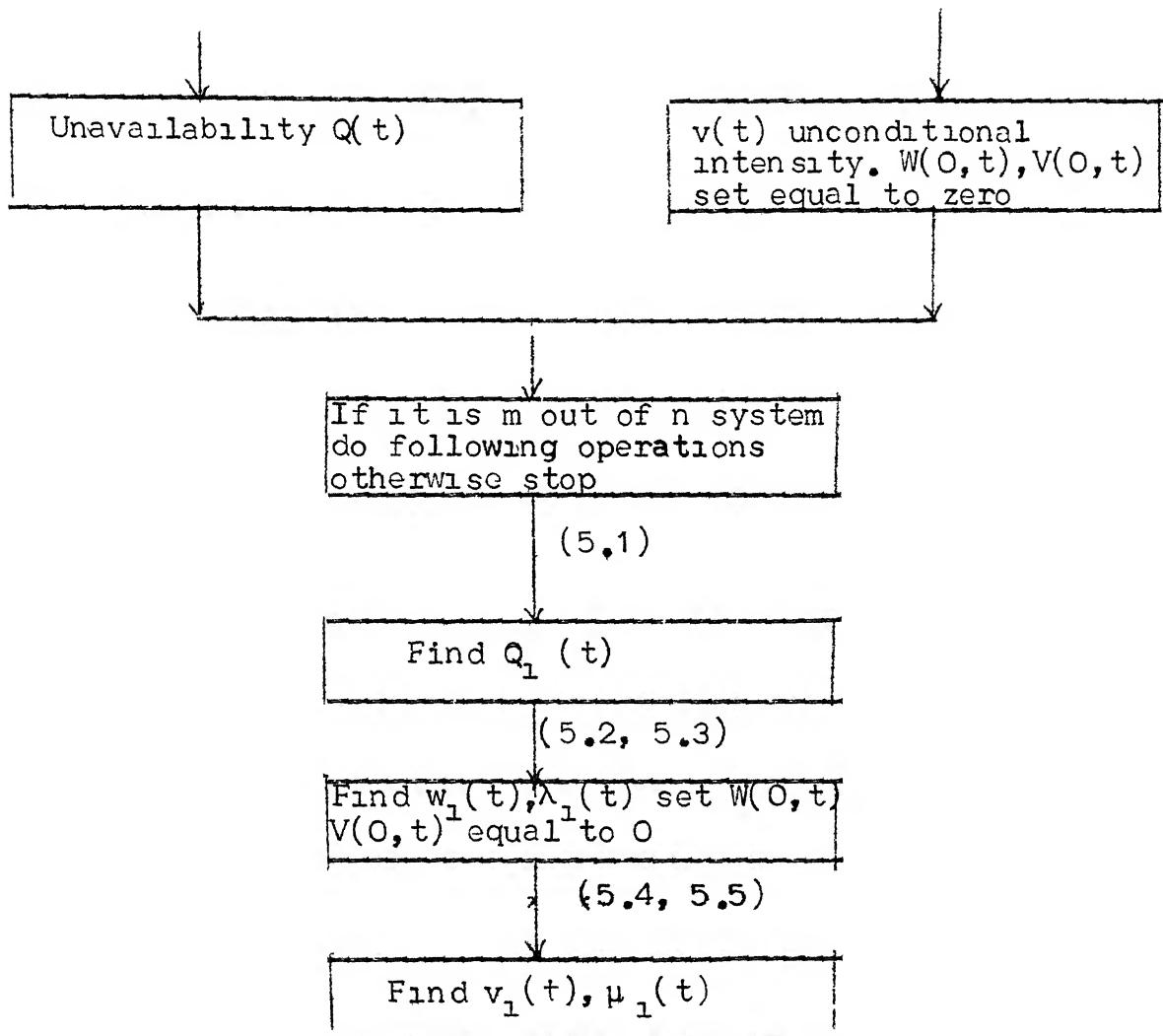
$$\begin{aligned} v_s(t) &= \binom{n}{m} m v (1-Q)^{m-1} \\ &= \frac{n!}{(m-1)! (n-m)!} (1-Q)^{m-1} \cdot v \end{aligned} \quad (5.4)$$

$$r_s(t) = \frac{v_s(t)}{Q_s(t)} \quad (5.5)$$

5.2 Flow Chart for System Analysis

Before looking at final flow charge, let's have a look on the component or basic event analysis.

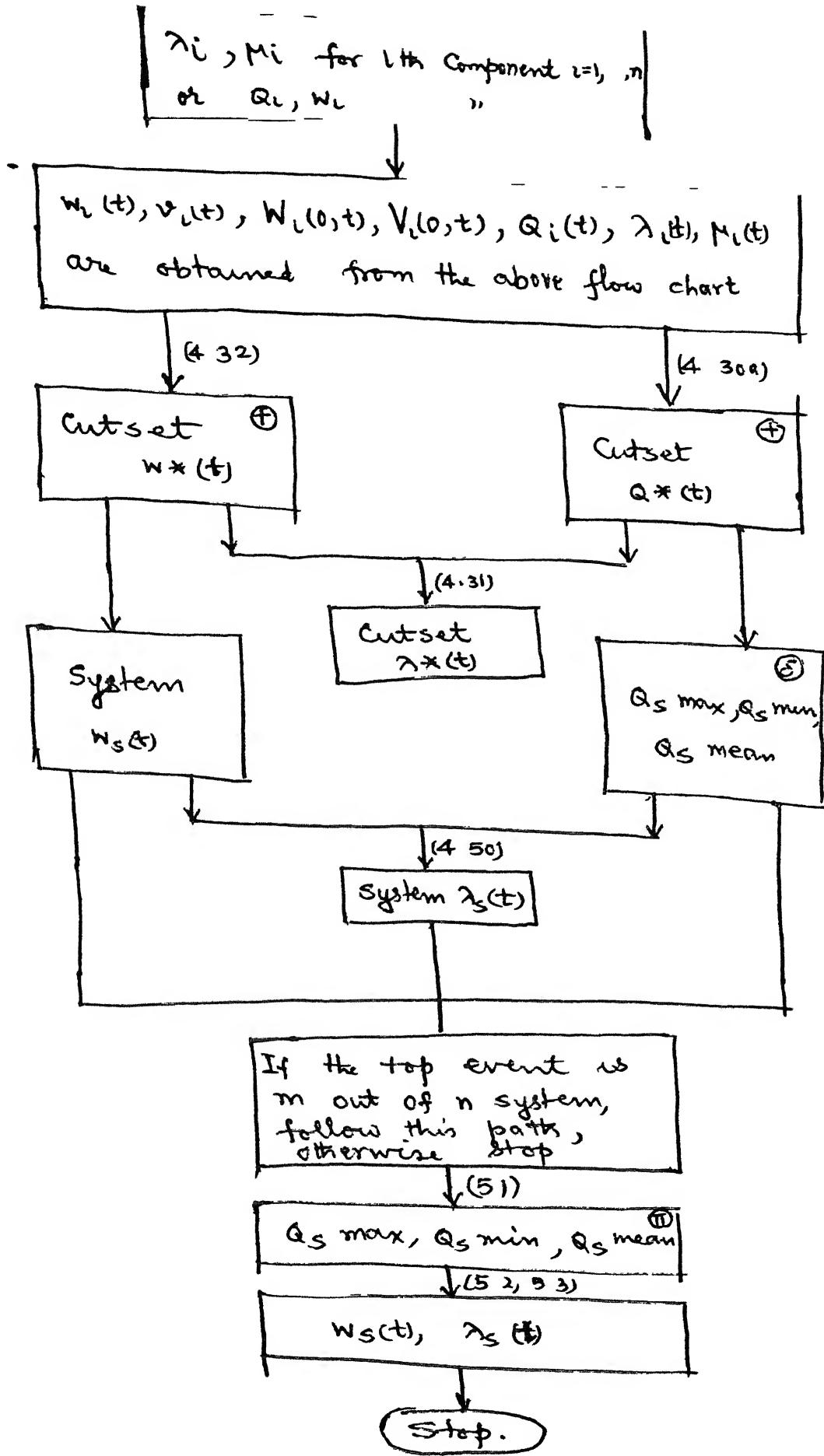




5.3 A Note on m Out of n System

If we do not define m out of n system as done in section 5.1 rather as — "Top event available if any of m or more systems out of n systems are available" — we can still use the formulation of (5.1) if we do the following. Replace m by $n-m+1$.

System analysis:



- (+) Not all cutsets are taken. There are 3 options:
 - Option 0 \Rightarrow Take all cutsets
 - Option 1 \Rightarrow Maximum order of cutsets to be considered are specified. Higher order cutset $Q_{s,j}$ are set to zero.
 - Option 2 Or more \Rightarrow % accuracy desired is specified.
If $\frac{Q_{\max}^* - Q^*}{Q_{\max}^*} > \% \text{ assigned}$
 Q^* is set to zero.
- (o) - Bracketting has been used to find $Q_{s,\max}$, $Q_{s,\min}$. If the first three terms taken in equation (4.36a), it gives $Q_{s,\max}$. If first two terms taken, eqn.(4.36a) gives $Q_{s,\min}$.

$$Q_s \text{ mean} = \sqrt{Q_{s,\max} * Q_{s,\min}} \quad (5.6)$$

This geometric mean has been suggested in WASH-1400 report [4,5].
- (π) - For all these quantities $Q_{s,\max}$, $Q_{s,\min}$, $Q_{s,\text{mean}}$, same equation (5.1) has been used. Hence $Q_{s,\max}$ and $Q_{s,\min}$ are not very accurate but $Q_{s,\text{mean}}$ is exact.

5.4 Input to Program

In file named FOR 24.DAT give:

In the first line - Time and decimal accuracy of result wanted.

In 2nd line onwards component no, $\lambda(Q)$, $\mu(w)$, M,N, MN for all basic events. M,N indicates M out of N system.

$$MN = 0 \Rightarrow P_1 = \lambda \quad P_2 = \mu$$

$$MN \neq 0 \Rightarrow P_1 = Q(T) \quad P_2 = w(T)$$

Next line must begin with 0 to indicate end of basic events.

Next line we have to give OPTION = ?

If Option = 0 \Rightarrow take all cutsets.

If option = 1 \Rightarrow Next line must contain

$$\text{MAX ORDER} = ?$$

If option $\geq 2 \Rightarrow$ Next line must contain % = ?

Next lines contain cutsets one in each line till all cutsets are over. End of cutset information again indicated by a line starting with 0 or a black line.

Last line has M and N for the TOP GATE.

5.5 Output

First few lines have $w(t)$, $v(t)$, $W(0,t)$, $V(0,t)$, Lambda, Meu, and $Q(t)$ for basic events.

Next we have

$w * (t)$, $\lambda * (t)$ and $Q * (t)$ for the relevant cutsets only.

Lastly, λ_s , $w(s)$, $Q(s)_{\min}$, $Q(s)_{\max}$, $Q(s)_{\text{mean}}$ for the total system.

Note: $W(0,t)$, $V(0,t)$ is set to zero for some basic events, this has been done to reduce computational effort. These quantities are required for calculation of $Q(t)$, so if $Q(t)$ is obtained by any alternate formulation or as such provided in input these quantities have no relevance and hence have been set to zero.

5.6 Merits of the Program

(i) In all formulation the most important quantity $Q(t)$ (unavailability) calculation has been done exactly except for bracketing. First three terms of equation (4.36a) has been taken for upper bound which make it more close to actual value. Even for M out of N system unavailability calculation is exact.

(ii) Fussell's approximation has been used for quantities which are not of primary interest like $w(t)$ of m out of n system and $w_s(t)$ for the entire system. Rest of the other quantities' formulation is exact like $W(0,t)$, $V(0,t)$, $w(t)$, $v(t)$ for components.

(iii) Answer accurate to any decimal place can be obtained. The iteration procedure and integration procedure has been written in a such a manner that it automatically reduces the step size of iteration for obtaining the desired

decimal place accuracy.

(iv) Only those cutsets have been considered whose reliability is large enough according to certain criterion enumerated in the form of OPTION. This reduces the computational effort drastically. This reduces the computational effort drastically. In the current input, only 12 out of total 84 cutsets are important for 99% accuracy.

5.7 Demerits and Suggestions for Further Improvement

(i) The bracketting procedure has not been kept flexible. The number of terms for upper bound approximation and lower bound approximation has not been kept variables but are 3 and 2 in this program. So desired accuracy of Q_{mean} can not be specified. This is not a particularly severe demerit since the successive terms of equation (4.36a) decrease by $Q(t)$ which is a small number and hence two bounds found above is almost close to the actual value . In any case Q_{mean} is found by a method suggested by WASH - 1400 report.Hence, it is very dependable .

(ii) Fussell's approximation has been used for some quantities which is valid only for large time. This again is not a severe constraint since the approximation has been used only for quantities of secondary importance. MTTF and MTTR are in a few hours generally less than 100 whereas the time to be considered for calculation is in years. Hence

condition of the kind equation (4.43) and large time is usually satisfied.

(iii) $v(t)$, $\mu(t)$ for cutsets and overall system has not been found. This has not been done in KITT either. The reason for this is that these quantities are not of primary interest. Secondly, not all components are repairable hence calculation in those cases is not necessary.

5.8 Discussion of Result

For LMFBR primary loop, we obtained the following result for unavailability of the system fault tree we started out with.

Time	$\wedge(s)$	$w(s)$	$\varrho(s)_{\min}$	$Q(s)_{\max}$	$Q(s)_{\text{mean}}$
1 year	0.5916×10^{-11}	0.5916×10^{-11}	0.25666×10^{-8}	0.257×10^{-8}	0.2569×10^{-8}
40 years	0.9374×10^{-10}	0.9374×10^{-10}	0.7857×10^{-7}	0.7938×10^{-7}	0.7898×10^{-7}

After some improvement in the system by increasing redundancies in components (in parallel) as in the print-out, we got the following:

Time	λ (s)	w(s)	$Q(s)_{min}$	$Q(s)_{max}$	$Q(s)_{mean}$
1 year	0.8637×10^{-13}	0.8637×10^{-13}	0.38851×10^{-10}	0.38868×10^{-10}	0.38859×10^{-10}
40 years	0.3425×10^{-10}	0.3425×10^{-10}	0.24798×10^{-10}	0.25016×10^{-7}	0.24907×10^{-7}

We observe the following facts from our analysis:

(i) With time, system unavailability increases. Even at the end of 40 years unavailability is 0.25×10^{-7} to 0.8×10^{-7} which is quite satisfactory.

(ii) As we increase redundancies, unavailability gets reduced. Our analysis can, thus, help in system design. We can decide the redundancies (standby and parallel components) based on some top event failure probability according to certain unavailability allocation techniques. Adequacy of the design can also be ascertained.

(iii) Even for 99% accuracy, not all minimum cutsets are important. So, it is better to concentrate on the important minimum cutsets rather than all. Also with time the important minimum cutsets keep on changing. Some old important cutsets no longer remain that important and some new adds to the list of important ones.

(iv) As the % accuracy required is decreased, not many new cutsets are added. So we can assign less accuracy

desired without much increase in error. Like, in place of 99%, we can assign 85%.

(v) Some components are more critical in determining the system unavailability. Change of redundancy of some component has much pronounced effect on the overall unavailability than others.

Our result is closely comparable to the value 4×10^{-7} per reactor year obtained by F.J. Baloh, N.W. Brown, J. Graham, A.M. Smith, P.P. Zemanick [8] for Clinch River Breeder reactor plant. The value is also consistent with the primary allocation (goal) set which was less than equal to 8×10^{-7} . This number was allocated on the basis of 10 CFR 100 criterion near plant site boundary [8].

CHAPTER 6

ILLUSTRATIVE PROGRAM FOR MINIMAL PATH AND CUTSET FOR AN ELECTRICAL SYSTEM

6.1 Introduction

Electrical systems differ from the basic nuclear system in that its fault tree has a network graph like structure. There are many feedback paths, parallel paths and interconnections. This is not like the fault tree containing OR and AND gate and hence a separate formulation is necessary for its analysis.

In the technique used in the program, we do not require all the minimal paths to be deduced and checked. A few paths (called basic minimal paths) are deduced from the minimal path tree. The combination of failures that breaks the set of basic minimal paths is sufficient to deduce all the minimal cutsets of the n/w. The set of basic minimal paths is a subset of all the minimal paths of the n/w and the remaining minimal paths are not necessary for evaluating the minimal cutsets [20].

6.2 Definition

Minimal Path - It is a path from source to sink whereby no nodes are traversed more than once [19].

Basic Minimal Path - It is a minimal path in which no element of the path are linked to another by any branch in the n/w except with those elements before or after it in the path.

6.3 Program Details

Program is closely based on [ao] algorithm details of which can be looked into. What we have done is an improvement in it to make it more efficient and doing minor corrections. The output format has been changed to make it more useful.

CHAPTER 7

FAULT TREE MODIFICATION PROGRAM (PROGRAM-4)

7.1 Object

Apart from AND and OR gates, there are many other gates which occur in the fault trees like XOR, NOT, NAND NOR, Priority AND, Inhibit etc. Priority AND and Inhibit gates are treated as AND gates. There should be some method to take care of XOR, NOT, NAND, NOR, M out of N gate. 'M' out of 'N' gate we took care at quantification level. The present program aims at taking care of XOR, NAND, NOR and NOT gates.

7.2 Difficulty

The gate equivalence for XOR leads to change in tree-level. Some method must be evolved to find sub-branch leading to change of level. Whole fault tree data must be modified accordingly. There are many zeros in some of basic event entries, these ^{also} must be taken care of.

7.3 Approach

XOR - gate has the following gate equivalence

$$A \oplus B = \bar{A} \cdot B + A \cdot \bar{B} \quad [1]$$

(See Figure 1)

NOT - gates are taken care of by following -

De Morgan's law -

$$(i) \overline{A + B} = \overline{A} \cdot \overline{B}$$

$$(ii) \overline{A \cdot B} = \overline{A} + \overline{B}$$

Example

See Figure 2

XOR gate is first replaced by combination of two AND and one OR gate. Negation at any gate is removed by changing AND to OR or OR to AND gate with input negated. This process continued till we come to negated basic events and tree containing only AND and OR gates.

7.4 Program Details

As earlier we denote AND gate by an even number and OR gate by an odd Number. XOR gate is indicated by an odd Number larger than 100. NAND, NOR are indicated by a negative even or odd integer respectively. Next, we follow the following procedure:

(i) First entry of each basic event input data is checked for XOR gate. If no XOR gate, we go to next basic event. Then the second entry so on.

(ii) If any basic event has XOR gate, level of the branch containing that basic event is found. Level of branch is the number of gates encountered when we traverse from top event to that basic event by minimal path. If level

of the branch is equal to level of tree, Level (L) of the tree is increased by one.

(iii) All the data right to the XOR gate is shifted by 1 with event number placed in array GN (I, LB + 1), where LB = level of the branch concerned.

(iv) GN (I, J) = 1 and GN (I, J+1) = 2 with GN(I,J+2) negated. 1 indicates OR gate and 2 AND gate.

NOTE: GN(I,J) = Gate no. for Ith basic event and at Jth level of the branch containing it.

(v) Next line consists of copy of GN(I,J) \forall J with GN(I, J+1) = 4 and GN (I,J+2) unnegated. Hence all the entries below Ith line is shifted down by one.

(vi) Next line is looked into. If it contains the same XOR gate at the same level from top event, we do the following:

(a) GN (I,J) = 1 and GN (I, J+1) = 2
with GN (I,J+2) unnegated. 1 indicates OR gate
and 2 and 4 AND gates.

(b) Next line copy of GN (I,J) \forall J.GN (I+1, J) = 1
and GN (I+1, J+1) = 4 with GN (I, J+2) negated.

(vii) This procedure is followed for all the basic event entry. If XOR gate different or at different level in the branch concerned, we go to step (ii).

C 721 A 98977

(viii) When all entries has been looked into a particular level from top event, we consider the gates at next level from top event.

(ix) After entire input file has been modified, we restructure the tree according to the highest level decided in considering entire input file. If level is L, basic event in each line is placed at $GN(I, L+1)$ and other entries are filled with zeroes suitably.

(x) Negation of a particular line is considered next. If any gate is negative then even is changed to odd and odd to even (AND to OR gate and OR to AND gate) and next non-zero entry is negated till we encounter the basic event number. This procedure is repeated for all lines in the input file.

7.5 Output

and

We get a fault tree containing only AND/OR gates

Some of the basic events are negated and some unnegated. The resultant fault tree can be easily fed to minimal cut-set finding program to obtain minimal cutsets. These cutsets information can be given to fault tree quantification program to obtain top event unavailability and other parameters. Our minimal cutset finding program PCONCP is such that actual gate number is unimportant as long as an even number indicates AND and an odd no indicates OR gate. Thus repetition of 1 and 2 as gate numbers is modified fault

tree does not create problems in further analysis.

Modified fault tree and actual tree data can be traced easily from the print out. Two examples has been shown in the print out.

FINAL WORDS

We have complete package of fault tree analysis for nuclear systems. These system can contain electrical system (e.g. scram signals). Only thing left is fault tree construction program. But normally manual construction is done. Once fault tree is available, it can be analysed for top event failure probability using our package. Tree can have all kinds of gates. Final analysis depends on availability of basic event failure data. Assumption of constant failure and maintenance rates may be irritating. General time dependent analysis could be carried out by suitably modifying the programs. But this kind of exercise is unrealistic, since there is dearth of failure data.

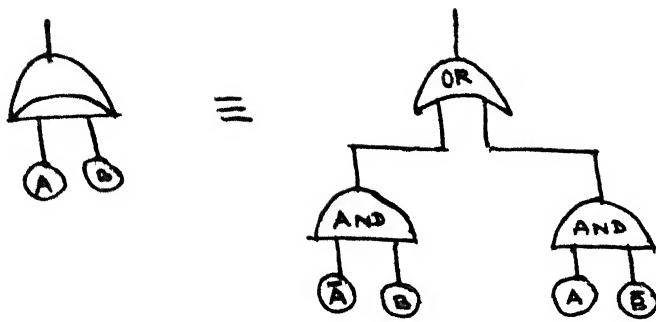


Figure 1 : Gate equivalence

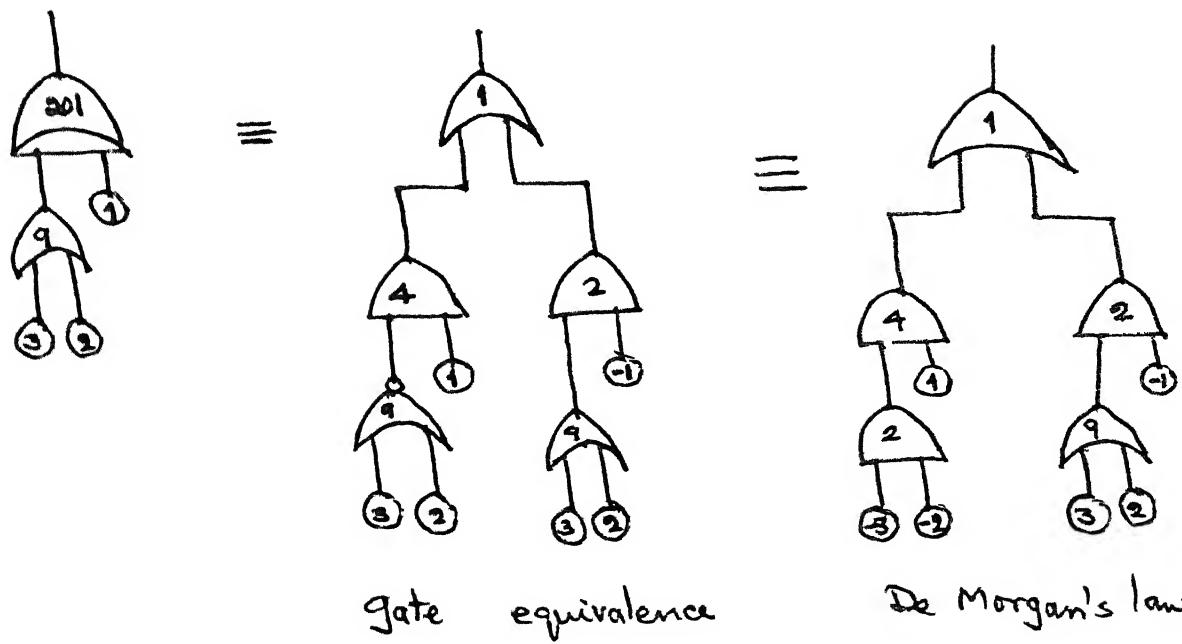


Figure 2 : Example and manual construction.

APPENDIX - IDATA TO BE USED

A. WASH-1400 [1, 2] report gives the following data:

(i) Pump

Mode	Q or λ	Median
(a) Failure to start	3×10^{-4} - $3 \times 10^{-3}/d$	$1 \times 10^{-3}/d$
(b) Failure to run given start in normal environment	3×10^{-6} - $3 \times 10^{-4}/hr$	$3 \times 10^{-5}/hr$

(ii) Motor Operated Valve

(a) Failure to operate	3×10^{-4} - $3 \times 10^{-3}/d$	$1 \times 10^{-3}/d$
(b) Failure to remain open	3×10^{-5} - $3 \times 10^{-4}/d$	$1 \times 10^{-4}/d$
(c) λ	1×10^{-7} - $1 \times 10^{-6}/hr$	$3 \times 10^{-7}/hr$
(d) Rupture λ	1×10^{-9} - $1 \times 10^{-7}/hr$	$1 \times 10^{-8}/hr$

(iii) Check Valve

(a) Failure to open	3×10^{-5} - $3 \times 10^{-4}/d$	$1 \times 10^{-4}/d$
(b) Internal Leakage	1×10^{-7} - $1 \times 10^{-6}/hr$	$3 \times 10^{-7}/hr$

(iv) Relief valve

(a) Failure to open	3×10^{-6} - $3 \times 10^{-5}/d$	$1 \times 10^{-5}/d$
(b) Premature open	3×10^{-6} - $3 \times 10^{-5}/hr$	$1 \times 10^{-5}/hr$

(v) Pipe

(a) $\underline{3''}$ dia rupture	$3 \times 10^{-11} - 3 \times 10^{-8}/\text{hr}$	$1 \times 10^{-9}/\text{hr}$
(b) $\underline{3''}$ dia	$3 \times 10^{-12} - 3 \times 10^{-9}/\text{hr}$	$1 \times 10^{-10}/\text{hr}$

(VI) MTTR for

Pump = 37 hr (typical)

Valve = 24 hr (typical)

B. Green and Bourne book gives the following data - [9]

(a) Pipe $\sim 0.2 \times 10^{-6} \text{ hr}^{-1}$ [Failure rate](b) Control valve $\sim 30 \times 10^{-6} \text{ hr}^{-1}$ [Failure rate](c) Solenoid valve $\sim 30 \times 10^{-6} \text{ hr}^{-1}$ [Failure rate]

A.H. Earl has given the following information about pickering A and Bruce A life time incapacity upto the end of 1981 (CANDU) [10]:

	Pickering A	Bruce A
Heat transport pump	0.2 (years)	0.2 (years)
Pressure tubes	4.9 (years)	0.3 (years)
Boilers	0.5 (years)	2.4 (years)
Turbines & Generators	5.8 (years)	6.6 (years)
Heat exchangers	0.9 (years)	0.0 (years)
Valves	0.4 (years)	0.0 (years)

J.R. Aupied and H. Procaccia have given the following data [ii] for valves:

System	No. of oper. hrs.	MTTR	Mean unavail. time	hr^{-1}	on demand failure
Pneumatic valves	700000	22	53	16×10^{-6}	$5 \times 10^{-3}/\text{d}$
Check valve	600000	39	54	8×10^{-6}	$0.6 \times 10^{-3}/\text{d}$
Large flow rate valves	350000	15	70	50×10^{-6}	$4.5 \times 10^{-3}/\text{d}$
Small flow rate valves	330000	8	49	54×10^{-6}	$1.7 \times 10^{-3}/\text{d}$

Steam valves

Safety relief valve (low pressure)	560000	34		59×10^{-6}	$2 \times 10^{-3}/\text{d}$
Safety relief valve (high pressure)	2.25×10^6	24		49×10^{-6}	$8 \times 10^{-3}/\text{d}$
Check valve	1.43×10^6	11		6×10^{-6}	$0.03 \times 10^{-3}/\text{d}$
Motor operated valve	1.425×10^6	20		0.7×10^{-6}	$0.7 \times 10^{-3}/\text{d}$
Pneumatic valve	340000	24		65×10^{-6}	$1 \times 10^{-3}/\text{d}$

From A.H. Earl's data []

for pump

unavailability = $\frac{2}{25} = 8 \times 10^{-3}$ at the end of one year.

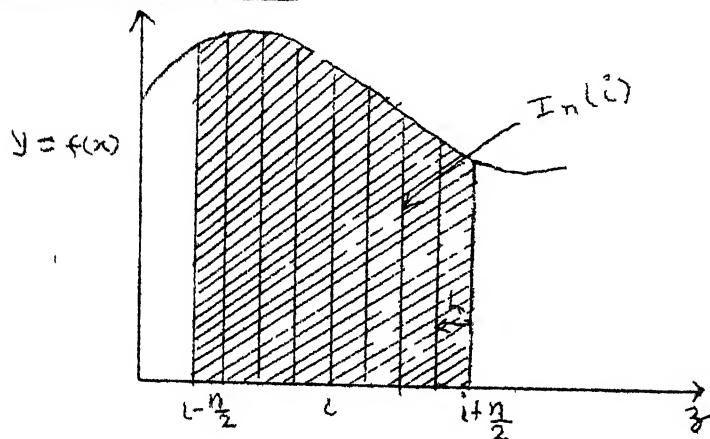
$$Q = 1 - e^{-\lambda t} \Rightarrow \lambda = \frac{1}{t} \ln \frac{1}{1-Q}$$

$$\therefore \lambda = 9.171 \times 10^{-7}$$

for Heat exchangers

$$Q = \frac{2}{25}$$

$$\lambda = 4.185 \times 10^{-6}$$

APPENDIX IIALGORITHM OF NUMERICAL INTEGRATION [15, 16]Simpson's Rule [15]

$$I_{2n}(z) = \frac{h}{3} (f_1 + 4f_2 + f_3) \quad (\text{II.1})$$

$$\text{Error} = -\frac{h^5}{90} f^{IV} \quad (\text{II.2})$$

$$S = \sum_{i=1,3,5,\dots,n-2} \frac{h}{3} (f_1 + 4f_2 + f_3) \quad (\text{II.3})$$

$$= \frac{h}{3} (f_1 + 4f_2 + 2f_3 + 4f_4 + 2f_5 + \dots + f_{n+1}) \quad (\text{II.4})$$

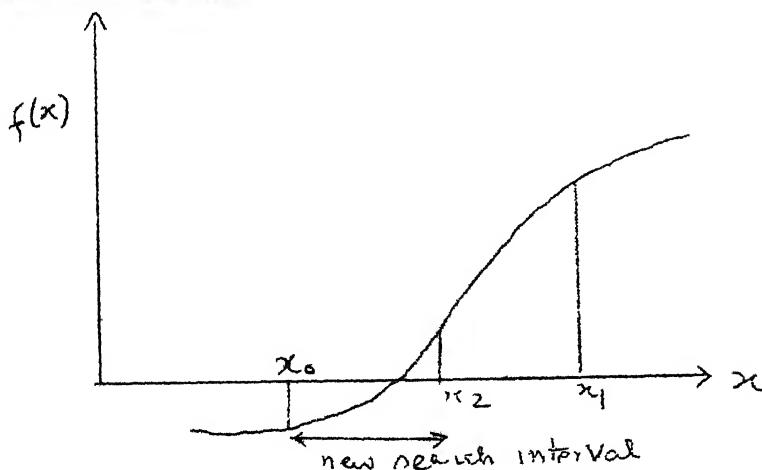
Hence f is to be tabulated at odd number of points.

If accuracy upto N th decimal place is desired

$$e = 0.5 * 10^{-N}$$

APPENDIX III
ITERATION TECHNIQUE

Bisection Method [15]



We begin by picking two trial points which enclose the root this is indicated by $f(x_0)$ and $f(x_1)$ being of opposite sign.

The interval (x_0, x_1) is bisected and mid point denoted by x_2 , i.e. $x_2 = (x_1 + x_0)/2$. If $f(x_2) = 0$ then x_2 is the root. If $f(x_2) > 0$ then root is between x_0 and x_2 . Hence replace x_1 by x_2 and search for root in this half interval.

If $f(x_2) < 0$ then root is between x_2 and x_1 . Hence replace x_0 by x_2 and again bisect the interval.

This bisection procedure is repeated till search interval is smaller than the precision with which answer is wanted.

Note that this method always encloses the root in the search interval and search interval is halved each time. Thus in 10 iterations, the search interval reduces by $2^{10} \approx 1000$ and in 20 by $10^{20} \approx 10^6$.

REFERENCES

1. WASH 1400: Reactor Safety Study, U.S., N.R.C., 1975, Appendix I.
2. WASH 1400. Reactor Safety Study, U.S., N.R.C., 1975, Appendix II.
3. "FETA - A Fast Fault Tree Analysis Program", T.Matsuoka, Nucl. Engg. and Design, Vol. 91, Jan. 1986, No.1, pp. 93-101.
4. "Efficient construction of Minimal Cutsets from Fault Trees" - S. Garrebba, P. Mussio, F. Naldi, G. Reina, G. Volta, IEEE Trans. on Reliability, Vol. R-26, No.2, June 1977, pp. 88-93.
5. "FATRAM - A core efficient cutset algorithm", D.M. Rasmuson, N.H. Marshall, IEEE Trans. on Reliability, Vol. R-27, No.4, Oct. 1978, pp. 250-253.
6. "AFTP - Fault Tree Analysis Program", R.A. Pullen, IEEE Trans. on Reliability, Vol. R-33, No.2, June 1984, pp. 171.
7. "The Heat Transfer System of SNR-300", S. Dreyer, Nucl. Engg. International, Vol. 21, No. 246, July 1976, pp. 49-50.
8. "Clinch River Breeder-Reactor Plant System Reliability", F.J. Baloh, N.W. Brown, J. Graham, A.M. Smith, P.P. Zemanick, IEEE Trans. on Reliability, Vol. R-25, No.3, Aug. 1976, pp. 191-196.
9. "Reliability Technology" - A.E. Green and A.J. Bourne, Wiley - Interscience, 1972.
10. "Reliability of CANDU Heat Transport Pump", A.H. Earl, Special Issue on the International Symposium on Reliability of Reactor Pressure Component, Vol. 81-82, pp. 85, March 1983.
11. "SRDF: A System for Collecting Reliability data from French PWR Power Plants. Method of Failure Analysis. Application to the Processing of Valve Data", J.R.Aupied and H. Procaccia, Vol. 81, No.1, 1984, Aug.(II), pp.127-137.

12. "Code of Federal Regulation, Title 10, "Energy"" , Jan. 1, 1975.
13. "Reliability Engineering and Risk Assessment" , E.J. Henley and H. Kumamoto, Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1981.
14. "Reliability Engineering" , A K. Govil, Tata McGraw-Hill Publishing Company Ltd., New Delhi, 1983.
15. "Computer Oriented Numerical Methods" - V. Rajaraman, Prentice Hall of India Private Ltd., New Delhi, 2nd Ed., 1984.
16. "Numerical Methods in FORTRAN" - J.M. McCormick, M.G. Salvadori, Prentice Hall of India Private Ltd., New Delhi, 1968.
17. "Approaches to Compacting Cutset Enumeration Algorithms" , IEEE Trans. on Reliability, Vol. R-28, 1979 April, pp. 62-63, by A. Rosenthal.
18. P.C. Williams, A.A. Esan, R.N. Adams, "Efficient Computation of Distribution System Reliability" , Proc. 6th Power System Computation, Conference, Damstadd, 1978.
19. M.O. Locks, "Relationship between Minimal Path Sets and Cutsets" , IEEE Trans. on Reliability, Vol. R-27, 1978, June, p. 106.
20. "A New Technique in Minimal Path and Cutset Evaluation" - G.B. Jasmon, O.S. Kai, IEEE Trans. on Reliability, Vol. R-24, No.2, June, 1985.

1

PCON version 102(2067) running C sequenced 5995 in stream 1
 Input from DSKC:C:CTL[15100,150064]
 Output to DSKC:C:LOG[15100,150064]
 No parameters

Time:00:01:00 Core:100P Unique:YES Restart:YES Output:VOUG

LOGIN 15100/150064 /DEFER/SPDOL:ALG/TIME:50/CDRE:100P/LOCATE:10/NAME:"MANDI KUMAR"
 DB 16 I I T KANPUR 603A(3) ITY116
 GNJSP Other jobs same PPN:18
 198 13-Dec-86 Sat
 You have some mail. Type *.BOX.

TY C, FOR
 4:59:01J

```

1000 3** Program for Minimal Cutsets using Prime Numbers
1200 C1 PROGRAM NAME "PCOMCP"
1400
1500 THIS PROGRAM USES PRIME NO.' CODING FOR BASIC EVENTS.
1600 AFTER MINIMIZATION DECODES THEM BACK.
1700 INPUT LINES AS MANY AS NO.' OF BASIC EVENTS ONLY.
1800 UPTO 100 BASIC INPUTS AND 100 OR GATES COULD BE ANALYSED..
1900 METHOD OF PROVIDING INPUT IS
2000 TO THE FIRST LINE GIVE LEVEL NO., NO. OF OR GATES & MAX
2200 ORDER OF MINIMAL CUTSET WANTED.
2300 LEVEL NO.' IS MAX. NO. OF GATES ENCOUNTERED IN COMING
2400 FROM TOP GATE TO ANY OF THE BASIC EVENTS.
2500 FROM NEXT LINE ONWARDS GIVE GATE NOS. ENCOUNTERED IN
2600 COMING FROM TOP TO ANY BASIC EVENT.
2700 LAST NO. OF THESE LINES IS BASIC EVENT NO.
2800 LAST LINE MUST BEGIN WITH 0 AND ARBITRARY OTHER VJS.
2900 ORDER OF BASIC EVENTS DOES NOT MATTER.
3000 OR GATES MUST HAVE ODD GATE NO.' AND AND GATE EVEN NO.'.
3100 INPUTS ARE TO BE GIVEN IN FILE NAME 'FOR20.DAT'.
3200 OUTPUT COMES IN FILE NAME 'FOR21.DAT'.
3300
3400 =====
3500 VARIABLE INDEX:=
3600 -----
3700 P - PRIME VOS.
3800 EN - EVENT VOS.
3900 LP - LEVEL POINTER
4000 ST - STORED VALUE
4100 DR - NO. OF OR GATES: VO. > THIS COULD ALSO BE GIVEN
4200 PF - PRIME FACTOR
4300 DIV - DIVIDER FOR FINDING PRIME NOS.
4400 S - STORAGE
4500 K - INTERMEDIATE STORAGE
4600 L - LEVEL
4700 MO - MAX ORDER UP TO WHICH CUTSETS ARE WANTED.
4800 P1F(100) - PRIME FACTOR ARRAY
4900 NG(100) - GATE NO. "
5000 N(100,100) - OPERATION "
5100 PC(100) - PRIME NOS. "
5200 PF(100) - PRIME FACTOR "
5300 S(100) - STORAGE "
5400
5500
5600
5700
5800
5900
6000
6100
6200
6300
6400
6500
6600
6700
6800
6900
7000
7100
7200
7300
7400
7500
7600
7700
7800
7900
8000
8100
8200
8300
8400
8500
8600
8700
8800
8900
9000
9100
9200
9300
9400
9500
9600
9700
9800
9900
10000
10100
10200
10300
10400
10500
10600
10700
10800
10900
11000
11100
11200
11300
11400
11500
11600
11700
11800
11900
12000
12100
12200
12300
12400
12500
12600
12700
12800
12900
13000
13100
13200
13300
13400
13500
13600
13700
13800
13900
14000
14100
14200
14300
14400
14500
14600
14700
14800
14900
15000
15100
15200
15300
15400
15500
15600
15700
15800
15900
16000
16100
16200
16300
16400
16500
16600
16700
16800
16900
17000
17100
17200
17300
17400
17500
17600
17700
17800
17900
18000
18100
18200
18300
18400
18500
18600
18700
18800
18900
19000
19100
19200
19300
19400
19500
19600
19700
19800
19900
20000
20100
20200
20300
20400
20500
20600
20700
20800
20900
21000
21100
21200
21300
21400
21500
21600
21700
21800
21900
22000
22100
22200
22300
22400
22500
22600
22700
22800
22900
23000
23100
23200
23300
23400
23500
23600
23700
23800
23900
24000
24100
24200
24300
24400
24500
24600
24700
24800
24900
25000
25100
25200
25300
25400
25500
25600
25700
25800
25900
26000
26100
26200
26300
26400
26500
26600
26700
26800
26900
27000
27100
27200
27300
27400
27500
27600
27700
27800
27900
28000
28100
28200
28300
28400
28500
28600
28700
28800
28900
29000
29100
29200
29300
29400
29500
29600
29700
29800
29900
30000
30100
30200
30300
30400
30500
30600
30700
30800
30900
31000
31100
31200
31300
31400
31500
31600
31700
31800
31900
32000
32100
32200
32300
32400
32500
32600
32700
32800
32900
33000
33100
33200
33300
33400
33500
33600
33700
33800
33900
34000
34100
34200
34300
34400
34500
34600
34700
34800
34900
35000
35100
35200
35300
35400
35500
35600
35700
35800
35900
36000
36100
36200
36300
36400
36500
36600
36700
36800
36900
37000
37100
37200
37300
37400
37500
37600
37700
37800
37900
38000
38100
38200
38300
38400
38500
38600
38700
38800
38900
39000
39100
39200
39300
39400
39500
39600
39700
39800
39900
40000
40100
40200
40300
40400
40500
40600
40700
40800
40900
41000
41100
41200
41300
41400
41500
41600
41700
41800
41900
42000
42100
42200
42300
42400
42500
42600
42700
42800
42900
43000
43100
43200
43300
43400
43500
43600
43700
43800
43900
44000
44100
44200
44300
44400
44500
44600
44700
44800
44900
45000
45100
45200
45300
45400
45500
45600
45700
45800
45900
46000
46100
46200
46300
46400
46500
46600
46700
46800
46900
47000
47100
47200
47300
47400
47500
47600
47700
47800
47900
48000
48100
48200
48300
48400
48500
48600
48700
48800
48900
49000
49100
49200
49300
49400
49500
49600
49700
49800
49900
50000
50100
50200
50300
50400
50500
50600
50700
50800
50900
51000
51100
51200
51300
51400
51500
51600
51700
51800
51900
52000
52100
52200
52300
52400
52500
52600
52700
52800
52900
53000
53100
53200
53300
53400
53500
53600
53700
53800
53900
54000
54100
54200
54300
54400
54500
54600
54700
54800
54900
55000
55100
55200
55300
55400
55500
55600
55700
55800
55900
56000
56100
56200
56300
56400
56500
56600
56700
56800
56900
57000
57100
57200
57300
57400
57500
57600
57700
57800
57900
58000
58100
58200
58300
58400
58500
58600
58700
58800
58900
58900
59000
59100
59200
59300
59400
59500
59600
59700
59800
59900
60000
60100
60200
60300
60400
60500
60600
60700
60800
60900
61000
61100
61200
61300
61400
61500
61600
61700
61800
61900
62000
62100
62200
62300
62400
62500
62600
62700
62800
62900
63000
63100
63200
63300
63400
63500
63600
63700
63800
63900
64000
64100
64200
64300
64400
64500
64600
64700
64800
64900
65000
65100
65200
65300
65400
65500
65600
65700
65800
65900
66000
66100
66200
66300
66400
66500
66600
66700
66800
66900
67000
67100
67200
67300
67400
67500
67600
67700
67800
67900
68000
68100
68200
68300
68400
68500
68600
68700
68800
68900
68900
69000
69100
69200
69300
69400
69500
69600
69700
69800
69900
70000
70100
70200
70300
70400
70500
70600
70700
70800
70900
71000
71100
71200
71300
71400
71500
71600
71700
71800
71900
72000
72100
72200
72300
72400
72500
72600
72700
72800
72900
73000
73100
73200
73300
73400
73500
73600
73700
73800
73900
73900
74000
74100
74200
74300
74400
74500
74600
74700
74800
74900
74900
75000
75100
75200
75300
75400
75500
75600
75700
75800
75900
75900
76000
76100
76200
76300
76400
76500
76600
76700
76800
76900
76900
77000
77100
77200
77300
77400
77500
77600
77700
77800
77900
77900
78000
78100
78200
78300
78400
78500
78600
78700
78800
78900
78900
79000
79100
79200
79300
79400
79500
79600
79700
79800
79900
79900
80000
80100
80200
80300
80400
80500
80600
80700
80800
80900
80900
81000
81100
81200
81300
81400
81500
81600
81700
81800
81900
81900
82000
82100
82200
82300
82400
82500
82600
82700
82800
82900
82900
83000
83100
83200
83300
83400
83500
83600
83700
83800
83900
83900
84000
84100
84200
84300
84400
84500
84600
84700
84800
84900
84900
85000
85100
85200
85300
85400
85500
85600
85700
85800
85900
85900
86000
86100
86200
86300
86400
86500
86600
86700
86800
86900
86900
87000
87100
87200
87300
87400
87500
87600
87700
87800
87900
87900
88000
88100
88200
88300
88400
88500
88600
88700
88800
88900
88900
89000
89100
89200
89300
89400
89500
89600
89700
89800
89900
89900
90000
90100
90200
90300
90400
90500
90600
90700
90800
90900
90900
91000
91100
91200
91300
91400
91500
91600
91700
91800
91900
91900
92000
92100
92200
92300
92400
92500
92600
92700
92800
92900
92900
93000
93100
93200
93300
93400
93500
93600
93700
93800
93900
93900
94000
94100
94200
94300
94400
94500
94600
94700
94800
94900
94900
95000
95100
95200
95300
95400
95500
95600
95700
95800
95900
95900
96000
96100
96200
96300
96400
96500
96600
96700
96800
96900
96900
97000
97100
97200
97300
97400
97500
97600
97700
97800
97900
97900
98000
98100
98200
98300
98400
98500
98600
98700
98800
98900
98900
99000
99100
99200
99300
99400
99500
99600
99700
99800
99900
99900
100000
100100
100200
100300
100400
100500
100600
100700
100800
100900
100900
101000
101100
101200
101300
101400
101500
101600
101700
101800
101900
101900
102000
102100
102200
102300
102400
102500
102600
102700
102800
102900
102900
103000
103100
103200
103300
103400
103500
103600
103700
103800
103900
103900
104000
104100
104200
104300
104400
104500
104600
104700
104800
104900
104900
105000
105100
105200
105300
105400
105500
105600
105700
105800
105900
105900
106000
106100
106200
106300
106400
106500
106600
106700
106800
106900
106900
107000
107100
107200
107300
107400
107500
107600
107700
107800
107900
107900
108000
108100
108200
108300
108400
108500
108600
108700
108800
108900
108900
109000
109100
109200
109300
109400
109500
109600
109700
109800
109800
109900
109900
110000
110100
110200
110300
110400
110500
110600
110700
110800
110900
110900
111000
111100
111200
111300
111400
111500
111600
111700
111800
111900
111900
112000
112100
112200
112300
112400
112500
112600
112700
112800
112900
112900
113000
113100
113200
113300
113400
113500
113600
113700
113800
113900
113900
114000
114100
114200
114300
114400
114500
114600
114700
114800
114900
114900
115000
115100
115200
115300
115400
115500
115600
115700
115800
115900
115900
116000
116100
116200
116300
116400
116500
116600
116700
116800
116900
116900
117000
117100
117200
117300
117400
117500
117600
117700
117800
117800
117900
118000
118100
118200
118300
118400
118500
118600
118700
118800
118800
118900
119000
119100
119200
119300
119400
119500
119600
119700
119800
119800
119900
119900
120000
120100
120200
120300
120400
120500
120600
120700
120800
120900
120900
121000
121100
121200
121300
121400
121500
121600
121700
121800
121900
121900
122000
122100
122200
122300
122400
122500
122600
122700
122800
122900
122900
123000
123100
123200
123300
123400
123500
123600
123700
123800
123900
123900
124000
124100
124200
124300
124400
124500
124600
124700
124800
124900
124900
125000
125100
125200
125300
125400
125500
125600
125700
125800
125900
125900
126000
126100
126200
126300
126400
126500
126600
126700
126800
126900
126900
127000
127100
127200
127300
127400
127500
127600
127700
127800
127800
127900
128000
128100
128200
128300
128400
128500
128600
128700
128800
128800
128900
129000
129100
129200
129300
129400
129500
129600
129700
129800
129800
129900
129900
130000
130100
130200
130300
130400
130500
130600
130700
130800
130900
130900
131000
131100
131200
131300
131400
131500
131600
131700
131800
131900
131900
132000
132100
132200
132300
132400
132500
132600
132700
132800
132900
132900
133000
133100
133200
133300
133400
133500
133600
133700
133800
133900
133900
134000
134100
134200
134300
134400
134500
134600
134700
134800
134900
134900
135000
135100
135200
135300
135400
135500
135600
135700
135800
135900
135900
136000
136100
136200
136300
136400
136500
136600
136700
136800
136900
136900
137000
137100
137200
137300
137400
137500
137600
137700
137800
137800
137900
138000
138100
138200
138300
138400
138500
138600
138700
138800
138800
138900
139000
139100
139200
139300
139400
139500
139600
139700
139800
139800
139900
139900
140000
140100
140200
140300
140400
140500
140600
140700
140800
140900
140900
141000
141100
141200
141300
141400
141500
141600
141700
141800
141900
141900
142000
142100
142200
142300
142400
142500
142600
142700
142800
142900
142900
143000
143100
143200
143300
143400
143500
143600
143700
143800
143900
143900
144000
144100
144200
144300
144400
144500
144600
144700
144800
144900
144900
145000
145100
145200
145300
145400
145500
145600
145700
145800
145900
145900
146000
146100
146200
146300
146400
146500
146600
146700
146800
146900
146900
147000
147100
147200
147300
147400
147500
147600
147700
147800
147800
147900
148000
148100
148200
148300
148400
148500
148600
148700
148800
148800
148900
149000
149100
149200
149300
149400
149500
149600
149700
149800
149800
149900
149900
150000
150100
150200
150300
150400
150500
150600
150700
150800
150900
150900
151000
151100
151200
151300
151400
151500
151600
151700
151800
151900
151900
152000
152100
152200
152300
152400
152500
152600
152700
152800
152900
152900
153000
153100
153200
153300
153400
153500
153600
153700
153800
153900
153900
154000
154100
154200
154300
154400
154500
154600
154700
154800
154900
154900
155000
155100
155200
155300
155400
155500
155600
155700
155800
155900
155900
156000
156100
156200
156300
156400
156500
156600
156700
156800
156900
156900
157000
157100
157200
157300
157400
157500
157600
157700
157800
157800
157900
158000
158100
158200
158300
158400
158500
158600
158700
158800
158800
158900
159000
159100
159200
159300
159400
159500
159600
159700
159800
159800
159900
159900
160000
160100
160200
160300
160400
160500
160600
160700
160800
160900
160900
161000
161100
161200
161300
161400
161500
161600
161700
161800
161900
161900
162000
162100
162200
162300
162400
162500
162600
162700
162800
162900
162900
163000
163100
163200
163300
163400
163500
163600
163700
163800
163900
163900
164000
164100
164200
164300
164400
164500
164
```

```

1800 1 FORMAT(1X,3(I3,1X))
1900 C P(1)=2
2000 NN IS CHECK NO.
2100 NN=2
2200 I=1
2300 4 IF(I.GT.99) GO TO 2
2400 NN=NN+1
2500 DIV=2
2600 5 IF(MOD(NN,DIV).EQ.0) GO TO 4
2700 DIV=DIV+1
2800 1 IF(DIV.NE.NN) GO TO 5
2900 I=I+1
3000 C PRIME NOS. STORED IN P(I)
3100 P(I)=NN
3200 GO TO 4
3300 2 CONTINUE
3400 DO 20 I=1,100
3500 NG(I)=0
3600 PF(I)=0
3700 P1F(I)=0
3800 DO 21 J=1,100
3900 NC(I,J)=0
4000 21 CONTINUE
4100 20 CONTINUE
4200 15 INITIALISATION
4300 READ(20,11),((NG(I),I=1,100,EV)
4400 FORMAT(1X,100(I3,1X))
4500 DO 7 I=1,100
4600 S(I)=0
4700 7 CONTINUE
4800 C1 IF(NG(1).EQ.0) GO TO 170
4900 CONDITION INDICATES END OF DATA
5000 DO 12 I=1,100
5100 IF(EN.NE.I) GO TO 12
5200 EN=P(I)
5300 GO TO 13
5400 12 CONTINUE
5500 13 LOOP FOR PRIME NO CODING
5600 LP=L
5700 DO 30 I=1,L
5800 IF(NG(LP).NE.0) GO TO 35
5900 LP=LP-1
6000 30 CONTINUE
6100 C1 ASSIGNMENT OF INITIAL LEVEL POINTER i.e. MAX NO. OF GATE
6200 FROM FDP
6300 GO TO 40
6400 36 DO 38 J=1,DR
6500 IF(N(LP,J).NE.0) GO TO 40
6600 38 CONTINUE
6700 N(LP,1)=EN
6800 GO TO 45
6900 40 IF(MOD(NG(LP),2).NE.0) GO TO 55
7000 C1 IF NO. EVEN IT IS AND GATE; IF ODD IT IS OR GATE
7100 DO 56 J=1,DR
7200 N(LP,J)=V(LP,J)*EN
7300 56 CONTINUE
7400 C1 MULTIPLY (IF AND GATE) ALL THE ELEMENTS IN ARRAY
7500 GO TO 80
7600 55 DO 58 J=1,DR
7700 IF(N(LP,J).NE.0) GO TO 50
7800 GO TO 55
7900 50 CONTINUE
8000 N(LP,J)=EN

```

```

11100      50      STORE IF OR GATE IN ARRAY
11200      IF(CDP.EQ.1) GO TO 45
11300      LP=LP+1
11400      DECREASE LEVEL POINTER
11500      DO 85 J=1,DR
11600      IF(N(LP,I).EQ.0) GO TO 30
11700      GO TO 90
11800      CONTINUE
11900      DO 95 I=1,DR
12000      N(LP,I)=N(LP+1,I)
12100      N(LP+1,I)=0
12200      CONTINUE
12300      SHIFT IF N(I,I) NOT VACANT TO N(I-1,I)
12400      GO TO 45
12500      IF(MOD(N(LP),2).EQ.0) GO TO 100
12600      CHECK EVEN OR ODD
12700      DO 110 J=1,DR
12800      IF(N(LP,J).NE.0) GO TO 110
12900      GO TO 120
13000      CONTINUE
13100      K=0
13200      DO 130 I=J,100
13300      K=K+1
13400      IF(N(LP+1,K).EQ.0) GO TO 30
13500      N(LP,I)=N(LP+1,K)
13600      N(LP+1,K)=0
13700      CONTINUE
13800      SHIFTING OPERATION
13900      GO TO 80
14000      DO 131 I=1,DR
14100      IF(N(LP,I).EQ.0) GOTO 135
14200      SC(I)=N(LP,I)
14300      N(LP,I)=0
14400      CONTINUE
14500      K=0
14600      LOOP=I-1
14700      DO 140 I=1,LOOP
14800      DO 150 J=1,DR
14900      IF(K.GT.100) GOTO 160
15000      LOOP=SC(I)*N(LP+1,J)
15100      IF(LOOP.EQ.0) GOTO 140
15200      K=K+1
15300      N(LP,K)=LOOP
15400      CONTINUE
15500      CONTINUE
15600      DO 145 J=1,DR
15700      N(LP+1,J)=0
15800      CONTINUE
15900      INITIALIZING AFTER SHIFTING
16000      GO TO 80
16100      DO 180 I=1,100
16200      DO 190 J=1,100
16300      IF(I.EQ.J) GO TO 190
16400      IF(N(I,I).EQ.0) GOTO 180
16500      IF(N(1,J).EQ.0) GOTO 190
16600      IF((MOD(N(1,J),N(1,I))).EQ.0) N(1,J)=0
16700      CONTINUE
16800      180      CONTINUE
16900      190      WRITE(21,195)
17000      195      FORMAT(1X,'Following are the Minimal Cut Sets')
17100      DO 300 I=1,100
17200      DO 800 J=1,100
17300      PF(JI)=0

```

```

17400 P1F(JI)=0
17500 INITIALIZATION
17600 CONTINUE
17700 K=1
17800 I=1
17900 IF(T(1,I).EQ.0) GO TO 300
18000 IF(P(K).GT.N(1,I)) GO TO 290
18100 IF(MOD(N(1,I),P(K)).EQ.0) GO TO 250
18200 K=K+1
18300 270 DIFAVE OUT THOSE ELEMENTS IN N(I,I) WHICH ARE MULTIPLE
18400 OF EACH OTHER
18500 GO TO 220
18600 N(1,I)=N(1,I)/P(K)
18700 PRIME FACTORIZATION
18800 DO 260 J=1,100
18900 IF(P(J).NE.P(K)) GO TO 250
19000 PF(M)=J
19100 M=M+1
19200 GOTO 220
19300 CONTINUE
19400 GO TO 220
19500 DO 310 IN=1,99
19600 DO 320 J=IN+1,100
19700 IF(PF(J).EQ.PF(IN)) PF(J)=0
19800 320 CONTINUE
19900 310 CONTINUE
20000 K=0
20100 DO 265 J=1,100
20200 IF(PF(J).EQ.0) GO TO 265
20300 K=K+1
20400 P1F(K)=PF(J)
20500 P1F STORES THOSE FACTORS WHICH ARE NOT REPEATED
20600 IF(J.EQ.100) GO TO 280
20700 CONTINUE
20800 IF(MO.EQ.0) GO TO 280
20900 IF(K.GT.MO) GO TO 300
21000 280 WRITE(21,321),(P1F(IL0,IL=1,K)
21100 321 FORMAT(100I4))
21200 300 CONTINUE
21300 STOP
21400 END

```

TY FOR20.DAT
[14:09:04]

3	9		
1	2	0	2
1	2	5	1
1	4	0	3
1	4	3	4
1	5	0	5
0	5	0	6
			7
			8

EX C FOR
[14:09:04]
LINK: Loading
[LNKXCT C1 execution]

STOP

END OF EXECUTION

CPU TIME: 0.99 ELAPSED TIME: 0.78
EXIT

TY FDR21.DAT
[14:09:05]
Following are the Minimal Cut Sets
1 2 3
4 5
7 6
8

.KJDB/BATCH

[LGRAJL Another job is still logged-in under [15100,150064])
Job 16 Jser MANDJ KUMAR [15100,150064]
LOGged-off RTY116 at 14:09:05 on 13-Dec-86
Runtime: 0:00:01, KCS:19, Connect time: 0:00:10
Disk Reads:301, Writes:11
Batch version 102(2067) running C sequence 5998 in stream 1
Input from DSKC:C:CTL[15100,150064]
Output to DSKC:C:LOG[15100,150064]
Job parameters Time:00:01:00 Core:100P Unique:YES ReStart:YES Output:None

.LOGIN 15100/150064 /DEFER/SPJOL:ALL/TIME:50/CORE:100P/LOCATE:10/NAME:"MANDJ K
JDB 16 I I I KANPUR 603A(3) RTY116
[LOGNJSR Other jobs same PPN:18]
[410 13-Dec-86 Sat
You have some mail. Type *.BOX.

TY FDR22.DAT
[14:10:51]
7 25 5 7 0 0 0 1
2 1 5 7 0 0 0 2
2 1 5 7 0 0 0 3
2 2 1 5 7 0 0 4
2 2 1 5 7 0 0 5
2 2 1 5 7 0 0 6
2 2 1 5 7 0 0 7
2 2 1 5 7 0 0 8
2 2 1 5 7 0 0 9
2 2 1 5 7 0 0 10
2 2 1 5 7 0 0 11
2 2 1 5 7 0 0 12
2 2 1 5 7 0 0 13
2 2 1 5 7 0 0 14
2 2 1 5 7 0 0 15
2 2 1 5 7 0 0 16
2 2 1 5 7 0 0 17
2 2 1 5 7 0 0 18
2 2 1 5 7 0 0 19
2 2 1 5 7 0 0 20
2 2 1 5 7 0 0 21
2 2 1 5 7 0 0 22
2 2 1 5 7 0 0 23
2 2 1 5 7 0 0 24
2 2 1 5 7 0 0 25

EX C.FOR

[14:10:52]
LINK: Loading
[LVKXCT : execution]
STOP
END OF EXECUTION
CPU TIME: 1.87 ELAPSED TIME: 2.66
EXIT
TRY FOR21.DAT
[14:10:55]
Following are the Minimal Cut Sets
1 22
1 23
1 24
1 25
2 22
2 23
2 24
2 25
2 26
3 22
3 23
3 24
3 25
3 26
4 22
4 23
4 24
4 25
4 26
5 22
5 23
5 24
5 25
5 26
6 22
6 23
6 24
6 25
6 26
7 22
7 23
7 24
7 25
7 26
8 22
8 23
8 24
8 25
8 26
8 27
8 28
8 29
8 29
9 22
9 23
9 24
9 25
9 26
9 27
10 22
10 23
10 24
10 25
10 26
11 22
11 23
11 24
11 25
11 26
12 22
12 23
12 24
12 25
12 26
13 22
13 23

• KJCJ/BATCH

[UGTAJL] Another job is still logged-in under [15100,150064]
Job 16 User MANDU KUMAR [15100,150064]
Logged-off TTY115 at 14:10:56 on 13-Dec-13
Run time: 0:30:02, KCS:45, Connect time: 0:00:07
Disk Reads:2355, Writes:11

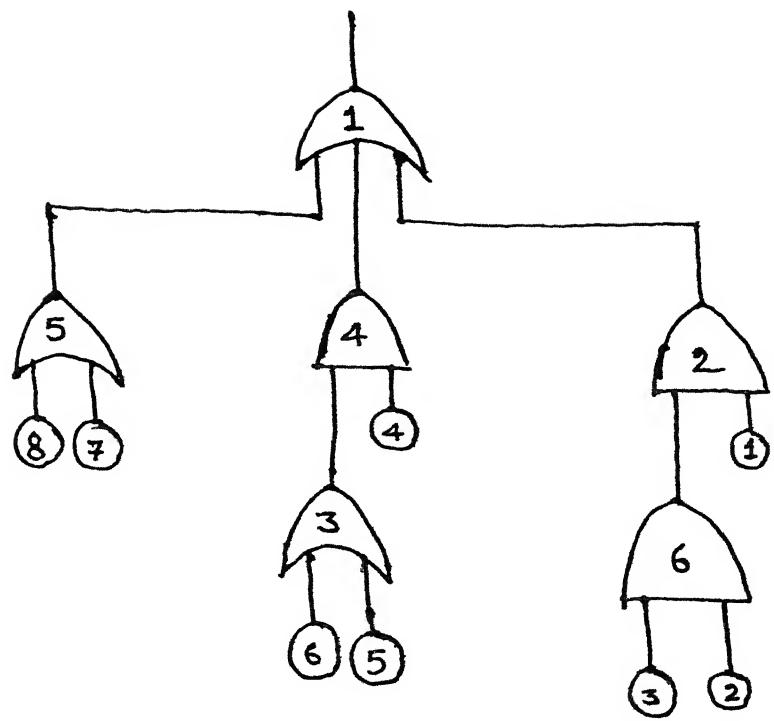


Figure : Example 1 for
PCOMCP

BATCON version 102(2067) running A sequencer 2752 in stream 1
 Input from DSKC:A:CTL[15100,150064]
 Output to DSKC:A:LDG[15100,150064]

JOB parameters

Time:00:01:00 Core:100P Unique:YES Restart:YES Output:VOLNG

LOGIN 15100/150064 /DEFER/SPJDL:ALL/TIMER:60/CORE:100P/LOCATE:10/NAME:"MANDI KUMAR"
 JOB 23 I 1 R KANPUR 603A(3) TTY115
 [LGNJSP, Other jobs same PPN:27]
 1340 29-Nov-86 Sat

TY A FOR1
 13:40:55J

00100
 00200
 00300
 00400
 00500
 00600
 00700
 00800
 00900
 01000
 01100
 01200
 01300
 01400
 01500
 01600
 01700
 01800
 01900
 02000
 02100
 02200
 02300
 02400
 02500
 02600
 02700
 02800
 02900
 03000
 03100
 03200
 03300
 03400
 03500
 03600
 03700
 03800
 03900
 04000
 04100
 04200
 04300
 04400
 04500
 04600
 04700
 04800
 04900
 05000
 05100
 05200
 05300
 05400
 05500
 05600
 05700
 05800
 05900
 06000
 06100
 06200
 06300
 06400
 06500
 06600
 06700
 06800
 06900
 07000
 07100
 07200
 07300
 07400
 07500
 07600
 07700
 07800
 07900
 08000
 08100
 08200
 08300
 08400
 08500
 08600
 08700
 08800
 08900
 09000
 09100
 09200
 09300
 09400
 09500
 09600
 09700
 09800
 09900
 10000
 10100
 10200
 10300
 10400
 10500
 10600
 10700
 10800
 10900
 11000
 11100
 11200
 11300
 11400
 11500
 11600
 11700
 11800
 11900
 12000
 12100
 12200
 12300
 12400
 12500
 12600
 12700
 12800
 12900
 13000
 13100
 13200
 13300
 13400
 13500
 13600
 13700
 13800
 13900
 14000
 14100
 14200
 14300
 14400
 14500
 14600
 14700
 14800
 14900
 15000
 15100
 15200
 15300
 15400
 15500
 15600
 15700
 15800
 15900
 16000
 16100
 16200
 16300
 16400
 16500
 16600
 16700
 16800
 16900
 17000
 17100
 17200
 17300
 17400
 17500
 17600
 17700
 17800
 17900
 18000
 18100
 18200
 18300
 18400
 18500
 18600
 18700
 18800
 18900
 19000
 19100
 19200
 19300
 19400
 19500
 19600
 19700
 19800
 19900
 20000
 20100
 20200
 20300
 20400
 20500
 20600
 20700
 20800
 20900
 21000
 21100
 21200
 21300
 21400
 21500
 21600
 21700
 21800
 21900
 22000
 22100
 22200
 22300
 22400
 22500
 22600
 22700
 22800
 22900
 23000
 23100
 23200
 23300
 23400
 23500
 23600
 23700
 23800
 23900
 24000
 24100
 24200
 24300
 24400
 24500
 24600
 24700
 24800
 24900
 25000
 25100
 25200
 25300
 25400
 25500
 25600
 25700
 25800
 25900
 26000
 26100
 26200
 26300
 26400
 26500
 26600
 26700
 26800
 26900
 27000
 27100
 27200
 27300
 27400
 27500
 27600
 27700
 27800
 27900
 28000
 28100
 28200
 28300
 28400
 28500
 28600
 28700
 28800
 28900
 29000
 29100
 29200
 29300
 29400
 29500
 29600
 29700
 29800
 29900
 30000
 30100
 30200
 30300
 30400
 30500
 30600
 30700
 30800
 30900
 31000
 31100
 31200
 31300
 31400
 31500
 31600
 31700
 31800
 31900
 32000
 32100
 32200
 32300
 32400
 32500
 32600
 32700
 32800
 32900
 33000
 33100
 33200
 33300
 33400
 33500
 33600
 33700
 33800
 33900
 34000
 34100
 34200
 34300
 34400
 34500
 34600
 34700
 34800
 34900
 35000
 35100
 35200
 35300
 35400
 35500
 35600
 35700
 35800
 35900
 36000
 36100
 36200
 36300
 36400
 36500
 36600
 36700
 36800
 36900
 37000
 37100
 37200
 37300
 37400
 37500
 37600
 37700
 37800
 37900
 38000
 38100
 38200
 38300
 38400
 38500
 38600
 38700
 38800
 38900
 39000
 39100
 39200
 39300
 39400
 39500
 39600
 39700
 39800
 39900
 40000
 40100
 40200
 40300
 40400
 40500
 40600
 40700
 40800
 40900
 41000
 41100
 41200
 41300
 41400
 41500
 41600
 41700
 41800
 41900
 42000
 42100
 42200
 42300
 42400
 42500
 42600
 42700
 42800
 42900
 43000
 43100
 43200
 43300
 43400
 43500
 43600
 43700
 43800
 43900
 44000
 44100
 44200
 44300
 44400
 44500
 44600
 44700
 44800
 44900
 45000
 45100
 45200
 45300
 45400
 45500
 45600
 45700
 45800
 45900
 46000
 46100
 46200
 46300
 46400
 46500
 46600
 46700
 46800
 46900
 47000
 47100
 47200
 47300
 47400
 47500
 47600
 47700
 47800
 47900
 48000
 48100
 48200
 48300
 48400
 48500
 48600
 48700
 48800
 48900
 49000
 49100
 49200
 49300
 49400
 49500
 49600
 49700
 49800
 49900
 50000
 50100
 50200
 50300
 50400
 50500
 50600
 50700
 50800
 50900
 51000
 51100
 51200
 51300
 51400
 51500
 51600
 51700
 51800
 51900
 52000
 52100
 52200
 52300
 52400
 52500
 52600
 52700
 52800
 52900
 53000
 53100
 53200
 53300
 53400
 53500
 53600
 53700
 53800
 53900
 54000
 54100
 54200
 54300
 54400
 54500
 54600
 54700
 54800
 54900
 55000
 55100
 55200
 55300
 55400
 55500
 55600
 55700
 55800
 55900
 56000
 56100
 56200
 56300
 56400
 56500
 56600
 56700
 56800
 56900
 57000
 57100
 57200
 57300
 57400
 57500
 57600
 57700
 57800
 57900
 58000
 58100
 58200
 58300
 58400
 58500
 58600
 58700
 58800
 58900
 59000
 59100
 59200
 59300
 59400
 59500
 59600
 59700
 59800
 59900
 60000
 60100
 60200
 60300
 60400
 60500
 60600
 60700
 60800
 60900
 61000
 61100
 61200
 61300
 61400
 61500
 61600
 61700
 61800
 61900
 62000
 62100
 62200
 62300
 62400
 62500
 62600
 62700
 62800
 62900
 63000
 63100
 63200
 63300
 63400
 63500
 63600
 63700
 63800
 63900
 64000
 64100
 64200
 64300
 64400
 64500
 64600
 64700
 64800
 64900
 65000
 65100
 65200
 65300
 65400
 65500
 65600
 65700
 65800
 65900
 66000
 66100
 66200
 66300
 66400
 66500
 66600
 66700
 66800
 66900
 67000
 67100
 67200
 67300
 67400
 67500
 67600
 67700
 67800
 67900
 68000
 68100
 68200
 68300
 68400
 68500
 68600
 68700
 68800
 68900
 69000
 69100
 69200
 69300
 69400
 69500
 69600
 69700
 69800
 69900
 70000
 70100
 70200
 70300
 70400
 70500
 70600
 70700
 70800
 70900
 71000
 71100
 71200
 71300
 71400
 71500
 71600
 71700
 71800
 71900
 72000
 72100
 72200
 72300
 72400
 72500
 72600
 72700
 72800
 72900
 73000
 73100
 73200
 73300
 73400
 73500
 73600
 73700
 73800
 73900
 74000
 74100
 74200
 74300
 74400
 74500
 74600
 74700
 74800
 74900
 75000
 75100
 75200
 75300
 75400
 75500
 75600
 75700
 75800
 75900
 76000
 76100
 76200
 76300
 76400
 76500
 76600
 76700
 76800
 76900
 77000
 77100
 77200
 77300
 77400
 77500
 77600
 77700
 77800
 77900
 78000
 78100
 78200
 78300
 78400
 78500
 78600
 78700
 78800
 78900
 79000
 79100
 79200
 79300
 79400
 79500
 79600
 79700
 79800
 79900
 80000
 80100
 80200
 80300
 80400
 80500
 80600
 80700
 80800
 80900
 81000
 81100
 81200
 81300
 81400
 81500
 81600
 81700
 81800
 81900
 82000
 82100
 82200
 82300
 82400
 82500
 82600
 82700
 82800
 82900
 83000
 83100
 83200
 83300
 83400
 83500
 83600
 83700
 83800
 83900
 84000
 84100
 84200
 84300
 84400
 84500
 84600
 84700
 84800
 84900
 85000
 85100
 85200
 85300
 85400
 85500
 85600
 85700
 85800
 85900
 86000
 86100
 86200
 86300
 86400
 86500
 86600
 86700
 86800
 86900
 87000
 87100
 87200
 87300
 87400
 87500
 87600
 87700
 87800
 87900
 88000
 88100
 88200
 88300
 88400
 88500
 88600
 88700
 88800
 88900
 89000
 89100
 89200
 89300
 89400
 89500
 89600
 89700
 89800
 89900
 90000
 90100
 90200
 90300
 90400
 90500
 90600
 90700
 90800
 90900
 91000
 91100
 91200
 91300
 91400
 91500
 91600
 91700
 91800
 91900
 92000
 92100
 92200
 92300
 92400
 92500
 92600
 92700
 92800
 92900
 93000
 93100
 93200
 93300
 93400
 93500
 93600
 93700
 93800
 93900
 94000
 94100
 94200
 94300
 94400
 94500
 94600
 94700
 94800
 94900
 95000
 95100
 95200
 95300
 95400
 95500
 95600
 95700
 95800
 95900
 96000
 96100
 96200
 96300
 96400
 96500
 96600
 96700
 96800
 96900
 97000
 97100
 97200
 97300
 97400
 97500
 97600
 97700
 97800
 97900
 98000
 98100
 98200
 98300
 98400
 98500
 98600
 98700
 98800
 98900
 99000
 99100
 99200
 99300
 99400
 99500
 99600
 99700
 99800
 99900
 100000

LL - LOWER LIMIT OF UNAVAILABILITY
 UL - UPPER LIMIT OF Q
 CN - COMPONENT NO.
 LAMBDA - MEAN LAMBDA
 WM - MEAN W

05500
 05600
 05700
 05800
 05900
 06000
 06100
 06200
 06300
 06400
 06500
 06600
 06700
 06800
 06900
 07000
 07100
 07200
 07300
 07400
 07500
 07600
 07700
 07800
 07900
 08000
 08100
 08200
 08300
 08400
 08500
 08600
 08700
 08800
 08900
 09000
 09100
 09200
 09300
 09400
 09500
 09600
 09700
 09800
 09900
 10000
 10100
 10200
 10300
 10400
 10500
 10600
 10700
 10800
 10900
 11000
 11100

COUNT= COUNTER
 FACT= FACTORIAL
 CS(100,100)= CUTSET MATRIX
 CQ(100)= CUTSET Q
 CW(100)= CUTSET W
 KS(100)= CUTSET ORDER ARRAY
 CLAMB(100)= CUTSET LAMBDA
 SV(100)= V(t)
 F= 0 BY MARKOV'S FORMULATION
 F1= w(t) "
 F2= v(t) "
 P1= LAMBDA DR 0 AS INDICATED EARLIER
 P2= MEU DR W(F) "
 T= TIME
 N1= DECIMAL PLACE UP TO WHICH ACCURACY WANTED
 M,N= M OUT OF N SYSTEM
 MN= INDICATOR WHETHER P1 IS LAMBDA DR 0; P2 IS MEU DR N
 DEPENDING ON WHETHER IT IS 0 OR NONZERO
 S= STORAGE FOR Q

REAL LL,LAMBDA1,MEU,LAMBDA
 INTEGER CN,CS,OPTION,COUNT,FACT
 DIMENSION O(100),SW(100),CS(100,100),CQ(100),CW(100),KS(100),
 1CLAMB(100),SV(100)
 EXTERNAL F,F1,F2,FACT
 COMMON P1,P2
 READ(24,20)T,N1
 FORMAT(F8.1,1X,I2)
 WRITE(5,25)
 25 FORMAT(/1X,'COMP NO' T W(t) V(t)
 1 W(0,t) V(0,t) LAMBDA
 2MEU Q(t))
 DO 100 I=1,100
 READ(24,30)CN,P1,P2,M,N,MN
 FORMAT(1X,I3,2(1X,E10.3),3(1X,I2))
 IF(CN.EQ.0)GOTO 105
 IF(MN.NE.0)GOTO 35
 SW(CN)=F1(P1,P2,T)
 SV(CN)=F2(P1,P2,T)
 CALL INT(F1,0.,T,N1,N)
 CALL INT(F2,0.,T,N1,V)
 O(CN)=W-V
 GOTO 38
 O(CN)=P1
 SW(CN)=P2
 LAMBDA=SW(CN)/(1.-Q(CN))
 P1=LAMBDA
 P2=0.
 CALL ET(F,Q(CN),N1)
 MEU=P2
 SV(CN)=F2(P1,P2,T)
 IF(M.EQ.0)GOTO 70
 ST=Q(CN)
 O(CN)=0.
 N2=N
 DO 40 J=M,N

3

11200
 11300
 11400
 11500
 11600
 11700
 11800
 11900
 12000
 12100
 12200
 12300
 12400
 12500
 12600
 12700
 12800
 12900
 13000
 13100
 13200
 13300

J2=J
 IV=FACT(N2)/(FACT(J2)*FACT(N2-J2))
 O(CN)=O(CN)+FLOAT(IV)*(ST**J2+((1.-ST)**(N2-J2))
 CONTINUE
 RT=FLOAT(FACT(N)/(FACT(N-1)*FACT(N-M)))
 SW(CN)=RT*(ST**((M-1))*SN(CN))
 SV(CN)=RT*((1.-ST)**((M-1))+SV(CN))
 MEU=SV(CN)/O(CN)
 LAMBDA=SW(CN)/(1.-Q(CN))
 W=0.
 V=0.
 GOTO 80
 LAMBDA=SW(CN)/(1.-Q(CN))
 MEU=SV(CN)/O(CN)
 WRITE(5,90),(CN,T,SW(I),SV(I),W,V,LAMBDA,MEU,Q(I))
 FORMAT(2X,I3,1X,F8.1,7(1X,E13.6))
 CONTINUE
 READ(24,110)OPTION
 FORMAT(7X,I1)
 IF(OPTION.EQ.0)GOTO 145
 IF(OPTION.GE.2)GOTO 130
 READ(24,120)MO

```

13400 120 FORMAT(10X,I2)
13500 PC=0.
13600 GOTO 143
13700 130 READ(24,140)PC
13800 140 FORMAT(2X,F4.1)
13900 145 MO=100
14000 143 WRITE(5,146)
14100 146 FORMAT(' //,1X,'CUT      t      w*(t)      Lambda*(t)      Q*(t)*
14200 150 COUNT=0
14300 DO 165 I=1,100
14400 READ(24,160),CS(I,I1),I1=1,100
14500 FORMAT(100(1X,I3))
14600 IF(CS(I,1).EQ.0.0)GOTO 170
14700 COUNT=COUNT+1
14800 165 CONTINUE
14900 170 SIMULATION OF CUTSET Q,CUTSET W & CUTSET LAMBDA EQUATIONS.
15000 DD 250 I=1,COUNT
15100 KS(I)=0
15200 CO(I)=1.
15300 CW(I)=0
15400 DO 200 I1=1,100
15500 IF(CS(I,I1).EQ.0.0)GOTO 205
15600 KS(I)=KS(I)+1
15700 200 CONTINUE
15800 205 DO 230 I1=1,KS(I)
15900 J=CS(I,I1)
16000 CQ(I)=CO(I)*Q(J)
16100 S=1.
16200 DO 220 L=1,KS(I)
16300 IF(J.EQ.CS(I,L))GOTO 220
16400 K=CS(I,L)
16500 S=S*D(K)
16600 220 CONTINUE
16700 CW(I)=CW(I)+SW(J)*S
16800 230 CONTINUE
16900 CLAMB(I)=CW(I)/(1.-CQ(I))
17000 250 CONTINUE
17100 AS PER OPTION ACTION IS TAKEN. IF OPTION=0, ALL CUTSETS TAKEN
17200 INTO ACCOUNT. IF OPTION=1, IF CUTSET ORDER GREATER THAN MAX ORDER
17300 SPECIFIED THEN Q IS REDUCED TO ZERO.
17400 IF OPTION>2 IF ACCURACY (MAX Q-Q)/MAX Q *100 >% ASSIGNED THEN

```

4

```

17500 C1 AGAIN Q REDUCED TO ZERO.
17600 IF(OPTION.EQ.0)GOTO 280
17700 TO FIND MAX CQ(I)=CN
17800 CV=CQ(1)
17900 DO 260 I=2,COUNT
18000 IF((CQ(I)).GT.CV)CV=CQ(I)
18100 CONTINUE
18200 DD 270 I=1,COUNT
18300 IF(OPTION.EQ.1)GOTO 262
18400 IF(((CV-CQ(I))*100./CV).GE.1.PC)GOTO 270
18500 GOTO 264
18600 262 IF(KS(I).LE.MD)GOTO 270
18700 CQ(I)=0.
18800 CLAMB(I)=0.
18900 CCW(I)=0.
19000 CONTINUE
19100 270 DD 285 I=1,COUNT
19200 IF(CQ(I).EQ.0.)GOTO 285
19300 WRITE(5,282)(I,I,CCW(I),CLAMB(I),CQ(I))
19400 FORMAT(1X,I2,1X,F8.4,3(1X,E13.6))
19500 285 CONTINUE
19600 C1 SIMULATION OF SYSTEM Q MAX,Q MIN,Q MEAN,LAMBDA,W(I)
19700 UL=0.
19800 C1 LOOP FINDS FIRST TERM OF SYSTEM Q EQUATION.
19900 DO 290 I=1,COUNT
20000 UL=UL+CQ(I)
20100 290 CONTINUE
20200 CDR1=0.
20300 C1 FINDS SECOND TERM OF SYSTEM Q EQUATION AS CDR1.
20400 DO 340 I=2,COUNT
20500 DO 330 J=1,I-1
20600 CUM=CQ(I)
20700 DO 320 K1=1,KS(J)
20800 DO 310 K2=1,KS(I)
20900 IF(CS(I,K2).EQ.CS(J,K1))GOTO 320
21000 340 CONTINUE
21100 KL=CS(J,K1)
21200 CUM=CUM+KL

```

```

21300 320 CONTINUE
21400 330 COR1=COR1+CUM
21500 340 CONTINUE
21600 350 CONTINUE
21700 360 LL=UL-COR1
21800 370 COR2=0
21900 380 FINDS THIRD TERM OF SYSTEM 2 EQUATION AS COR2.
22000 390 DO 400 I=3,COUNT
22100 400 DO 390 J=2,I-1
22200 410 DO 380 K=1,J-1
22300 420 CUM=CS(I)
22400 430 DO 380 K1=1,KS(J)
22500 440 DO 350 K2=1,KS(I)
22600 450 IF(CS(J,K1).EQ.CS(I,K2))GOTO 360
22700 460 CONTINUE
22800 470 KL=CS(J,K1)
22900 480 CUM=CUM*Q(KL)
23000 490 CONTINUE
23100 500 DO 378 K0=1,KS(K)
23200 510 DO 370 K2=1,KS(I)
23300 520 IF(CS(K,K0).EQ.CS(I,K2))GOTO 378
23400 530 CONTINUE
23500 540 DO 375 K1=1,KS(J)
23600 550 IF(CS(K,K0).EQ.CS(J,K1))GOTO 378
23700 560 CONTINUE

```

5

```

23800 570 KL=CS(K,K0)
23900 580 CUM=CUM*Q(KL)
24000 590 CONTINUE
24100 600 COR2=COR2+CUM
24200 610 CONTINUE
24300 620 CONTINUE
24400 630 CONTINUE
24500 640 UL=LL+COR2
24600 650 OM=SORT(UL*LL)
24700 660 READ(24,450)M,N
24800 670 FORMAT(2I3)
24900 680 IF(M.EQ.0)GOTO 465
25000 690 CI AS PER OUR FORMULATION , TO FIND Q FOR M OUT OF N SYSTEM
25100 700 ST=OM
25200 710 OM=0.
25300 720 ST1=UL
25400 730 ST2=LL
25500 740 UL=0.
25600 750 LL=0.
25700 760 DO 460 J=M,N
25800 770 J2=J
25900 780 N2=N
26000 790 IV=FACT(N2)/(FACT(J2)*FACT(N2-J2))
26100 800 OM=OM+FLOAT(IV)*(ST1**J2)*((1.-ST1)**(N2-J2))
26200 810 UL=UL+FLOAT(IV)*(ST1**J2)*((1.-ST1)**(N2-J2))
26300 820 LL=LL+FLOAT(IV)*(ST2**J2)*((1.-ST2)**(N2-J2))
26400 830 CONTINUE
26500 840 WM=0.
26600 850 LAMBDM=0.
26700 860 CI FUSSEL'S APPROXIMATION
26800 870 DO 470 I=1,COUNT
26900 880 WM=WM+CW(I)
27000 890 LAMBDM=LAMBDM+CLAMB(I)
27100 900 CONTINUE
27200 910 IF(M.EQ.0)GOTO 475
27300 920 RT=FLOAT(FACT(N)/(FACT(M-1)*FACT(N-M)))
27400 930 WM=RT*(ST1**M)*WM/(1.-(ST1**M))
27500 940 LAMBDM=WM/(1.-QM)
27600 950 WRITE(5,480)
27700 960 FORMAT(//,1X,'      t      LAMBDA(s)      w(s)      Q(s)MIN
27800 970 1. 0(S)MAX 0(S)MEAN   ')
27900 980 WRITE(5,490)(T,LAMBDM,WM,LL,UL,QM)
28000 990 FORMAT(1X,F8.1,5(1X,E13.6))
28100 1000 STOP
28200 1010 END
28300 1020 ****
28400 1030 NUMERICAL INTEGRATION
28500 1040 THIS SUBROUTINE USES SIMPSON FORMULA
28600 1050 X1, X2 ARE LOWER AND UPPER LIMITS OF INTEGRATION
28700 1060 N IS DECIMAL PLACE UPTO WHICH ACCURACY OF INTE. IS WANTED
28800 1070 OUTPUTS ALSO STEP USED FOR DESIRED ACCURACY
28900 1080 FUNCTION TO BE FED THROUGH AN EXTERNAL FUNCTION
29000 1090 F1 DEFINES THE FUNCTION

```

```

29200
29300      SUBROUTINE INT(F1,X1,X2,N,II)
29400      REAL I0,I1
29500      COMMON P1,P2
29600      H=(X2-X1)/2.
29700      I=2
29800      S1=F1(P1,P2,X1)+F1(P1,P2,X2)
29900      S2=0
30000      S4=F1(P1,P2,X1+H)

```

6

```

30100      I0=0
30200      I1=(S1+4.*S4)*(H/3.)
30300      IF(I1.EQ.0.0)GOTO 5
30400      10     IF(ABS((I1-I0)/I1).LE.0.5*10.**(-N))GOTO 5
30500      S2=S2+S4
30600      S4=0.
30700      X=X1+(H/2.)
30800      DO 2 J=1,I
30900      S4=S4+F1(P1,P2,X)
31000      X=X+H
31100      2:    CONTINUE
31200      H=H/2.
31300      I=2*I
31400      I0=I1
31500      I1=(S1+2.*S2+4.*S4)*(H/3.)
31600      GO TO 10
31700      5:    CONTINUE
31800      RETURN
31900      END
32000
32100      C1    ****
32200      FUNCTION F(P1,P2,T)
32300      IF(((P1+P2)*T).GT.60.)F=P1/(P1+P2)
32400      IF(((P1+P2)*T).LE.60.)F=(P1/(P1+P2))*(1.-EXP(-T*(P1+P2)))
32500      RETURN
32600      END
32700      C1    ****
32800
32900      C1    FUNCTION F1(P1,P2,T)
33000      IF(((P1+P2)*T).GT.60.)F1=P1*P2/(P1+P2)
33100      IF(((P1+P2)*T).LE.60.)F1=((P1*P2)+(P1**2)*EXP(-T*(P1+P2)))/(P1+P2)
33200      RETURN
33300      END
33400      C1    ****
33500
33600      C1    FUNCTION F2(P1,P2,T)
33700      IF(((P1+P2)*T).GT.60.)F2=P1*P2/(P1+P2)
33800      IF(((P1+P2)*T).LE.60.)F2=(P1*P2/(P1+P2))*(1.-EXP(-T*(P1+P2)))
33900      RETURN
34000      END
34100      C1    ****
34200
34300      INTEGER FUNCTION FACT(N)
34400      FACT=1
34500      IF(N.EQ.0)GOTO 1000
34600      DO 999 I=1,N
34700      FACT=FACT*I
34800      999   CONTINUE
34900      1000  RETURN
35000
35100      C1    !!!!!!!
35200
35300      SUBROUTINE ET(F,Q,N)
35400      COMMON P1,P2
35500      Y1=1.
35600      Y0=0.
35700      1050  IF(ABS((Y1-Y0)/Y1).LE.0.5*10.**(-N))GOTO 1500
35800      P2=(Y1+Y0)/2.
35900      Z=F(P1,P2,T)
36000      IF(Q-F(P1,P2,T)).LT.0.01300,1500,1400
36100      1300  Y1=B2
36200      1400  GOTO 1700
36300      Y0=P2

```

364000 1700 GOTO 1050
365000 1500 RETURN
366000 END

3 5

$\text{NCl}_3 = 7$
70.0

```

11 23
11 24
11 25
12 22
12 23
12 24
12 25
13 22
13 23
13 24
13 25
14 22
14 23
14 24
14 25
15 22
15 23
15 24
15 25
16 22
16 23
16 24
16 25
17 22
17 23
17 24
17 25
18 22
18 23
18 24
18 25
19 22
19 23
19 24
19 25
20 22
20 23
20 24
20 25
21 22
21 23
21 24
21 25
0 3
3 3

```

EX A F R
 [13:41 / 1]
 LINK: Loading
 [LVKXG, At execution]

9

COMP ID	t	w(t)	v(t)	W(0,t)	V(0,t)	LAMBDA	MU	Q(t)
1 350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-	
2 350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-	
3 350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-	
4 350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-	
5 350400.0	0.787257E-05	0.307515E-03	0.000000E+00	0.000000E+00	0.804236E-05	0.145657E-01	0.211123E-	
6 350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-	
7 350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-	
8 350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-	
9 350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-	
10 350400.0	0.681855E-07	0.830513E-05	0.000000E+00	0.000000E+00	0.633278E-07	0.398929E-02	0.208211E-	
11 350400.0	0.359118E-05	0.171662E-04	0.000000E+00	0.000000E+00	0.390094E-05	0.216182E-03	0.794055E-	
12 350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-	
13 350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-	

	14	350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-
	15	350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-
	16	350400.0	0.681823E-05	0.681774E-05	0.000000E+00	0.000000E+00	0.907488E-05	0.274167E-04	0.248571E+
	17	350400.0	0.297521E-04	0.297521E-03	0.000000E+00	0.000000E+00	0.311143E-04	0.679563E-02	0.437312E-
	18	350400.0	0.999965E-10	0.175044E-12	0.350394E-04	0.306769E-07	0.100000E-09	0.500002E-08	0.350087E-
	19	350400.0	0.100000E-11	0.140216E-15	0.350400E-06	0.245587E-11	0.100000E-11	0.400164E-10	0.350397E-
	20	350400.0	0.570749E-05	0.356718E-04	0.000000E+00	0.000000E+00	0.601946E-05	0.688290E-03	0.518267E-
	21	350400.0	0.570749E-05	0.356718E-04	0.000000E+00	0.000000E+00	0.601946E-05	0.688290E-03	0.518267E-
	22	350400.0	0.298211E-05	0.298211E-05	0.404497E+01	0.403900E+01	0.300001E-05	0.499579E-03	0.596305E-
	23	350400.0	0.199863E-08	0.823239E-10	0.700558E-03	0.445259E-04	0.200000E-08	0.120000E-06	0.686032E-
	24	350400.0	0.199986E-09	0.699550E-12	0.700775E-04	0.122634E-05	0.200000E-09	0.100000E-07	0.699549E-
	25	350400.0	0.247525E-05	0.247525E-05	0.857425E+00	0.857524E+00	0.250000E-05	0.249997E-03	0.990111E-

OUT t w*(t) Lambda*(t) p*(t)
 44 350400.0 0.232107E-06 0.232290E-06 0.785213E-03
 61 350400.0 0.782254E-06 0.783417E-06 0.148408E-02
 64 350400.0 0.683029E-06 0.584715E-06 0.245212E-02

t LAMBDA(S) S(S) Q(S)MIN Q(S)MAX Q(S)MEAN
 350400.0 0.937377E-10 0.937377E-10 0.785703E-07 0.793917E-07 0.789799E-07
 SJSP

END OF EXECUTION
 CPU TIME: 28.65 ELAPSED TIME: 2:38.02
 EXIT

.KJDB/BATCH

[LOGTAJL Another job is still logged-in under [15100,150064]
 Job 23 User MANDU KUMAR [15100,150064]
 Logged-off TTY115 at 13:43:45 on 29-Nov-86
 Runtime: 0:00:30, KCS:648, Connect time: 0:02:54
 Disk Reads:285, Writes:3
 BIPCJN Version 102(2067) running A sequential 2755 in stream 1
 Input from DSKC:A:CTL[15100,150064]
 Output to DSKC:A:LOG[15100,150064]
 Job parameters
 Line:00:01:00 Core:100P Unique:YES Restart:YES Output:VOLUG

.LOGIN 15100/150064 /DEFER/SJOB:ALL/TIMER:60/CDRE:100P/LOCATE:10/NAME:"MANDU KUMAR"
 JDB 23 I I T KANPUR 603A(3) TTY115
 [LGNJSP Other jobs same PPN:27]
 1346 29-Nov-86 Sat

.TY FOR24.DAT

10

[13:46:24]
 8760 0 5
 1 +1.0000E-10 +5.0000E-09
 2 +1.0000E-12 +4.0000E-11
 3 +1.0000E-10 +5.0000E-09
 4 +1.0000E-12 +4.0000E-11
 5 +1.5000E-05 +1.0000E-04 3 5
 6 +1.0000E-10 +5.0000E-09
 7 +1.0000E-12 +4.0000E-11
 8 +1.0000E-10 +5.0000E-09
 9 +1.0000E-12 +4.0000E-11
 10 +9.1700E-07 1.0000E-05 3 4
 11 +4.1840E-06 2.0000E-05 2 3
 12 +1.0000E-10 +5.0000E-09
 13 +1.0000E-12 +4.0000E-11
 14 +1.0000E-10 +5.0000E-09
 15 +1.0000E-12 +4.0000E-11
 16 +2.5000E-06 2.5000E-05 1 3
 17 +3.0000E-05 3.0000E-04 2 4

18 1.000E-10
19 1.000E-11
20 1.000E-10
21 1.000E-05
22 1.000E-06
23 1.000E-07
24 1.000E-08
25 1.000E-09

0 NCIIDC
10.0
1 22
1 23
1 24
1 25
2 22
2 23
2 24
2 25
3 22
3 23
3 24
3 25
4 22
4 23
4 24
4 25
5 22
5 23
5 24
5 25
6 22
7 23
7 24
7 25
8 22
9 23

22 33

11

9 23
9 24
9 25
10 22
10 23
10 24
10 25
11 22
11 23
11 24
11 25
12 22
12 23
12 24
12 25
13 22
13 23
13 24
13 25
14 22
14 23
14 24
14 25
14 22
15 23
15 24
15 25
16 22
16 23
16 24
16 25
17 22
17 23
17 24
17 25

18 22
 18 23
 18 24
 18 25
 19 22
 19 23
 19 24
 19 25
 20 22
 20 23
 20 24
 20 25
 21 22
 21 23
 21 24
 21 25
 0 3
 3 3

EX A FOR

[13:46:26]
LINK: Loading
[LNKXCR A execution]

IDMP NO	t	w(t)	v(t)	w(0,t)	v(0,t)	LAMBDA	MEU	Q(t)
1	8760.0	0.999999E-10	0.438050E-14	0.875000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-1
2	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-1
3	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-1

12

4	8760.0	0.100000E-11	0.356174E-13	0.875000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-1
5	8760.0	0.339019E-05	0.219585E-03	0.000000E+00	0.000000E+00	0.341048E-05	0.369218E-01	0.594731E-1
6	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-1
7	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-1
8	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-1
9	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-1
10	8760.0	0.640849E-09	0.905262E-05	0.000000E+00	0.000000E+00	0.640850E-09	0.506290E+00	0.478303E-1
11	8760.0	0.801790E-06	0.383265E-05	0.000000E+00	0.000000E+00	0.804355E-06	0.119740E-02	0.320381E-1
12	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-1
13	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-1
14	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-1
15	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-1
16	8760.0	0.735404E-05	0.145963E-05	0.000000E+00	0.000000E+00	0.730068E-05	0.254929E-04	0.572565E-1
17	8760.0	0.282559E-04	0.282559E-03	0.000000E+00	0.000000E+00	0.294128E-04	0.718401E-02	0.393317E-1
18	8760.0	0.999999E-10	0.438050E-14	0.876000E-06	0.191373E-10	0.100000E-09	0.500069E-08	0.875980E-1
19	8760.0	0.100000E-11	0.356174E-13	0.876000E-08	0.156580E-14	0.100000E-11	0.406592E-10	0.876000E-1
20	8760.0	0.249245E-05	0.155778E-04	0.000000E+00	0.000000E+00	0.251438E-05	0.178541E-02	0.872507E-1
21	8760.0	0.249245E-05	0.155778E-04	0.000000E+00	0.000000E+00	0.251438E-05	0.178541E-02	0.872507E-1
22	8760.0	0.298233E-05	0.294572E-05	0.261584E-01	0.202669E-01	0.300000E-05	0.500000E-03	0.589145E-1
23	8760.0	0.199996E-08	0.210128E-14	0.475198E-04	0.920530E-08	0.200000E-08	0.120000E-06	0.175106E-1
24	8760.0	0.200000E-09	0.175206E-13	0.175200E-05	0.767417E-10	0.200000E-09	0.400000E-07	0.175102E-1
25	8760.0	0.247796E-05	0.220423E-05	0.217705E-01	0.129536E-01	0.250000E-05	0.250000E-03	0.881691E-1

UP	t	w*(t)	Lambda*(t)	o*(t)

8760.0 0.214083E-06 0.214156E-06 0.837324E-03
8760.0 0.206719E-06 0.206823E-06 0.654825E-03
8760.0 0.283768E-06 0.283834E-06 0.231721E-03
8760.0 0.346592E-06 0.346712E-06 0.345784E-03

t LAMBDA(S) N(S) P(S)MIN P(S)MAX P(S)MEAN
50.0 0.591583E-11 0.591583E-11 0.255654E-08 0.257222E-08 0.255943E-08

OF EXECUTION
TIME: 39.83 ELAPSED TIME: 2:36.74

B/BATCH

AJL Another job is still logged-in under [15100,150064]
23 User MANOJ KUMAR [15100,150064]
ed-off TTY115 at 13:49:07 on 29-Nov-85
line: 0:00:40, KCS:887, Connect time: 0:02:45
Reads:217, Writes:3


```

00
01      SUBROUTINE BCJT
02      INTEGER I,J,K,L,M,N,NP,LP
03      COMMON /PATH/ MPT(25,40), MPJ(25,40), MPTD(25,40), MP(100,15), NP, LP
04      COMMON /CC/ 7,500, INC(50), ICOUNT(50)
05      DIMENSION LFIRST(20), LSE(40,2), LTHIRD(50,3), LFOUR(80,4)
06      DIMENSION LTC(15), INP(30,5), INC(50), MT(25), LM(45), B2I(15,20)
07      DIMENSION LBI(25,40)
08      -----FIRST ORDER CUTSET-----+
09
10      N1=0
11      NT=0
12      DO 20 I=1,LP
13      IF(ICOUNT(MP(NP,I)),EQ,NP) GOTO 10
14      NT=NT+1
15      INC(MP(NP,I))=-NT
16      LTC(NT)=MP(NP,I)
17      GOTO 20
18      INC(MP(NP,I))=-20
19      N1=N1+1
20      LFIRST(N1)=MP(NP,I)
21      CONTINUE
22      WRITE(5,200)(LFIRST(K), K=1,N1)
23      FORMAT(IX,' FIRST ORDER CUTSET : ',20I3)
24      DO 50 I=1,NP
25      NI=0
26      NO=0
27      NO=NO+1
28      IF(CMP(I,NO).EQ.0) GOTO 43
29      IF(CINC(MP(I,NO)).LT.0) GOTO 44
30      NI=NI+1
31      MT(NI)=MP(I,NO)
32      INC(MPCI,NO)=1
33      MPCI,NO)=0
34      GOTO 40
35      IF(CINC(MP(I,NO)),EQ,-20) GOTO 45
36      INC(I,INC(MP(I,NO))+1)=1
37      MPCI,NO)=0
38      GOTO 40
39      DO 50 K=1,NI
40      MP(I,K)=MT(K)
41      CONTINUE
42      NM=0
43      DO 55 J=1,NT
44      NM=NM+1
45      LM(NM)=LT(J)
46      DO 60 J=1,50
47      IF(CINC(J).NE.1) GOTO 60
48      NM=NM+1
49      LM(NM)=J
50      CONTINUE
51      IF(NP.LT.2) GOTO 130
52      -----SECOND ORDER CUTSET-----+
53
54      N2=0
55      DO 80 I=1,NT
56      NL2=0
57      DO 80 J=I+1,NM
58      IF(LT(I).EQ.LM(J)) GOTO 80
59      IF(ICOUNT(LT(I))+ICOUNT(LM(J)).LT.NP) GOTO 80
60      DO 75 K=1,NP
61      M=0
62      IF(INP(K,I).EQ.1) GOTO 75
63      IF(INC(LM(J)).GT.0) GOTO 70

```

```

17400
17500
17600
17700
17800
17900
18000
18100
18200
18300
18400
18500
18600
18700
18800
18900
19000
19100
19200
19300
19400
19500
19600
19700
19800
19900
20000
20100
20200
20300
20400
20500
20600
20700
20800
20900
21000
21100
21200
21300
21400
21500
21600
21700
21800
21900
22000
22100
22200
22300
22400
22500
22600
22700
22800
22900
23000
23100
23200
23300
23400
23500
23600

    IF(INP(K,INC(LM(J))*-1).EQ.1)GOTO 75
    M=M+1
    IF(MP(K,M).EQ.0)GOTO 80
    IF(MP(K,M).EQ.04(J))GOTO 75
    GOTO 70
  70  CONTINUE
    N2=N2+1
    LSEC(N2,1)=LT(I)
    LSEC(N2,2)=LM(J)
    NL2=NL2+1
    L2I(I,NL2)=LM(J)
  75  CONTINUE
    IF(N2.EQ.0)GOTO 95
    WRITE(5,201)
    FORMAT(1X,'SECOND ORDER CUTSET : ')
    DO 90 LL=1,N2
    WRITE(5,202)LL,LSEC(LL,1),LSEC(LL,2)
    FORMAT(14X,I3,':',3I4)
    IF(NP.LE.3)GOTO 130
  201  -----THIRD ORDER CUTSET-----+
    N3=0
    DO 110 I=1,NT
    NL3=0
    DO 110 J=I+1,NT
    DO 110 K=J+1,NT
    IF(ICOUNT(LT(I))+ICOUNT(LM(J))+ICOUNT(LM(K)).LT.NP)GOTO 140
    N=0
    N=N+1
    IF(L2I(I,N).EQ.0)GOTO 101
    IF(L2I(I,N).EQ.LM(J).OR.L2I(I,N).EQ.LM(K))GOTO 110
    GOTO 100
  100  IF(J.GT.NT)GOTO 102
    NMT=0
    NMT=NMT+1
    IF(L2I(J,NMT).EQ.0)GOTO 102
    IF(L2I(J,NMT).EQ.LM(K))GOTO 140
    GOTO 99
  99   DO 105 L=1,NT
    IF(INP(L,I).EQ.1)GOTO 105
    M=0
    IF(J.GT.NT)GOTO 103
    IF(INP(L,INC(LM(J))*-1).EQ.1)GOTO 105
    IF(K.GT.NT)GOTO 104
    IF(INP(L,INC(LM(K))*-1).EQ.1)GOTO 105
    M=M+1
    IF(MP(L,M).EQ.0)GOTO 110
    IF(MP(L,M).EQ.LM(J).OR.MP(L,M).EQ.LM(K))GOTO 105
    GOTO 104
  104  CONTINUE
    N3=N3+1
    LTHIRD(N3,1)=LT(I)
    LTHIRD(N3,2)=LM(J)
    LTHIRD(N3,3)=LM(K)
    NL3=NL3+1
    L3I(I,NL3)=LM(J)*100+LM(K)
  105  CONTINUE
    IF(N3.EQ.0)GOTO 130
    WRITE(5,203)
    FORMAT(1X,'THIRD ORDER CUTSET : ')
    DO 120 ML=1,N3
    WRITE(5,204)ML,(LTHIRD(ML,KL0,KL=1,3)
    FORMAT(14X,I3,':',3I4)
  130  RETURN

```

23700 END

TY FOR 22 DAT
14:38:31

12

EX.CR.FDR
[14:38:31J
LINK: Loading
[LINKCR CR execution]

SOURCE NODES: 12

FOR SINK NODE: 1

THE BASIC MINIMAL PATHS ::

S F D P

END OF EXECUTION
CPU TIME: 0.20 ELAPSED TIME: 1.18
EXIT

KJD8/BATCH

[LGFATJL] Another job is still logged-in under [15100,1500641]
Job 31 User MANDI KUMAR [15100,150054]
Logged-off ITY116 at 14:38:34 on 13-Dec-95
Runtime: 0:00:01, KCS:16, Connect time: 0:00:09

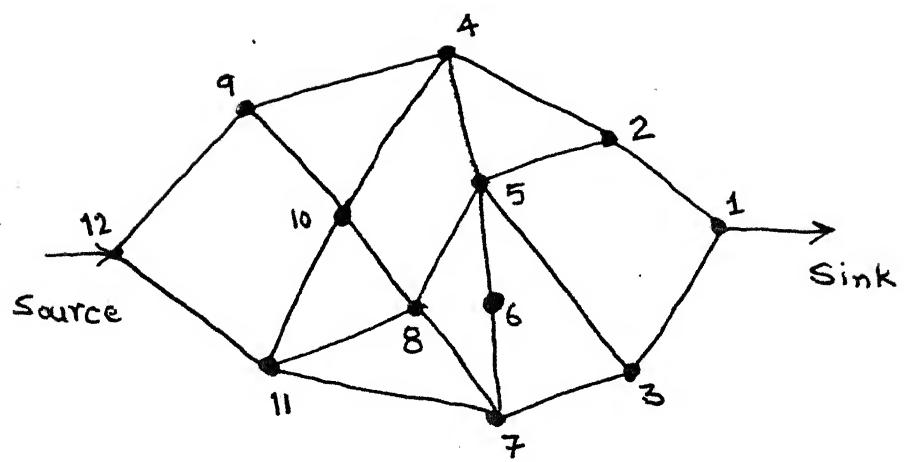


Figure: Example for MINCELS

ICON version 102(2067) running LAST sequence 2445 in stream 1
 Input from DSKC:LAST.CIL[15100,150064]
 Input to DSKC:LAST.LOG[15100,150064]
 No parameters
 Time:00:01:00 Core:100P Unique:YES Restart:YES Output:NOLUG

LOGIN 15100/150064 /DEFER/SPJDL:ALL/TIME:60/CODE:100P/LOCATE:10/NAME:"MANOJ KUMAR"
 DB 17 I I I KANPUR 603A(3) TTY115
 LGN JSP Other jobs same PPN:241
 323 29-Nov-86 Sat

TY LAST FOR
 3:23:24

```

0010 #####PROGRAM NAME: "M00TREE"
0020 #####THIS PROGRAM MODIFIES THE FAULT TREE TO TAKE CARE OF
0030 #####XOR AND NOT GATES. XOR GATE IS SHOWN BY ASSIGNING TO
0040 #####THE GATE NO. LARGER THAN 100. NOT GATE IS INPUTTED BY
0050 #####ASSIGNING IT A NEGATIVE NO. THUS, NOR AND NAND GATES
0060 #####ARE ALSO TAKEN CARE OF. OUTPUT CONSISTS OF TREE HAVING
0070 #####ONLY OR AND GATES AND BASIC EVENTS EITHER NEGATED
0080 #####OR UN-NEGATED.
0090 #####M OUT OF N KIND OF GATES ARE TAKEN CARE OF AT QUANTIFICATION STAGE.
0100 #####THE PROGRAM UTILISES DE MORGAN'S LAW AND GATE EQUIVALENCE.
0110 #####A+B=A.B + A.B=A.B
0120 #####***** ****
0130 #####INTEGER CK,GN(100,100),S,B,AD,DR
0140 #####L=LEVEL OF TREE, DR=NO. OF DR GATES IN THE TREE
0150 #####MO=MAX ORDER OF CUTSET IS WANTED.
0160 #####READ(22,1)D,DR,MD
0170 #####FORMAT(1X,3(13,1X))
0180 #####B=0
0190 #####INPUT FOR FAULT TREE
0200 #####DO 10 I=1,100
0210 #####READ(22,2),(GN(I,J),J=1,L+1)
0220 #####FFORMAT(100I4)
0230 #####IF(GN(I,1).EQ.0)GOTO 20
0240 #####B=B+1
0250 #####CONTINUE
0260 #####B=NO. OF BASIC EVENTS.
0270 #####S=0
0280 #####JP=0
0290 #####LS=L
0300 #####DO 130 J=1,100
0310 #####DO 140 I=1,100
0320 #####IF(I.GT.B)GOTO 130
0330 #####IF(J.GT.L)GOTO 135
0340 #####IF(GN(I,J).GT.100)GOTO 145
0350 #####DO 25 J1=J,L
0360 #####IF(GN(I,J1).EQ.0)GOTO 23
0370 #####AO=AO+1
0380 #####CONTINUE
0390 #####AO=LEVEL OF SUB-BRANCH CONTAINING THE BASIC EVENT.
0400 #####IF(AO.NE.L)GOTO 29
0410 #####LS=L
0420 #####L=L+1
0430 #####IF(GN(I,J).EQ.JP)GOTO 32
0440 #####S=0
0450
0460
0470
0480

```

```

00490      JP=GN(I,J)
00500      IF(S.NE.0) GOTO 35
00510      GK=GN(I,J+1)
00520      IP1=I+1
00530      IP2=I+2
00540      JP1=J+1
00550      JP2=J+2
00560      LP1=L+1
00570      GN(I,J)=1
00580      IF(GK.NE.GN(I,JP1)) GOTO 40
00590      IF(GN(I,LS).EQ.0) LPP=0
00600      IF(GN(I,LS).NE.0) LPP=L+1
00610      DO 80 J2=J,LPP
00620      M=LPP-J2+J
00630      IF(M.LT.JP2) GOTO 82
00640      GN(I,M)=GN(I,M-1)
00650      CONTINUE
00660      32
00670      GN(I,J+1)=2
00680      GN(I,JP2)=-1*GN(I,JP2)
00690      B=B+1
00700      DO 170 I2=IP2,B
00710      DO 160 J2=1,LP1
00720      M=B-I2+IP2
00730      GN(M,J2)=GN(M-1,J2)
00740      CONTINUE
00750      170
00760      CONTINUE
00770      DO 175 J2=1,LP1
00780      GN(IP1,J2)=GN(I,J2)
00790      CONTINUE
00800      GN(IP1,JP1)=4
00810      GN(IP1,JP2)=-1*GN(IP1,JP2)
00820      S=1
00830      GOTO 145
00840      IF(GN(I,LS).EQ.0) LPP=0
00850      IF(GN(I,LS).NE.0) LPP=L+1
00860      DO 45 J2=J,LPP
00870      M=LPP-J2+J
00880      IF(M.LT.JP2) GOTO 48
00890      GN(I,M)=GN(I,M-1)
00900      CONTINUE
00910      45
00920      GN(I,J+1)=2
00930      B=B+1
00940      DO 70 I2=IP2,B
00950      DO 60 J2=1,LP1
00960      M=B-I2+IP2
00970      GN(M,J2)=GN(M-1,J2)
00980      CONTINUE
00990      70
01000      DO 75 J2=1,LP1
01010      GN(IP1,J2)=GN(I,J2)
01020      CONTINUE
01030      75
01040      GN(IP1,JP1)=4
01050      GN(IP1,JP2)=-1*GN(IP1,JP2)
01060      S=1
01070      CONTINUE
01080      145
01090      CONTINUE
01100      140
01110      130
01120      135
01130      RESTRUCTURING OF TREE TO MAX ORDER AND LEVEL OF MODIFIED
01140      TREE
01150      DO 300 I=1,B
01160      DO 310 J=1,L
01170      IF(GN(I,L+1).NE.0) GOTO 300
01180      J1=L-J+1

```

```

01120 IF(GN(I,J1).EQ.0)GOTO 310.
01130
01140
01150
01160
01170      310  GN(I,L+1)=GN(I,J1)
01180      300  GN(I,J1)=0
01190      CONTINUE
01200      CONTINUE
01210      FOLLOWING TAKES CARE OF NOR, NDE, NAND GATES.
01220      DO 200 I=1,B
01230      DO 210 J=1,L
01240      IF(GN(I,J).GE.0)GOTO 210
01250      IF(CMOD(GN(I,J),2).NE.0)GOTO 230
01260      GN(I,J)=1
01270      GN(I,J+1)=-1*GN(I,J+1)
01280      GOTO 210
01290      GN(I,J)=2
01300      GN(I,J+1)=-1*GN(I,J+1)
01310      CONTINUE
01320      CONTINUE
01330      OUTPUTTING THE MODIFIED FAULT TREE.
01340      WRITE(5,240)L,OR,MJ
01350      240  FORMAT(IX,3(I3,1X))
01360      DO 260 I=1,B
01370      ARITE(5,250),(GN(I,J),J=1,L+1)
01380      FORMAT(1X,100I4)
01390      CONTINUE
01400      STOP
01410      END

```

:TY FOR22.DAT

[3:23:27]

1	1	10
210	1	
210	2	
0		

:EX LAST.FOR

[3:23:28]

LINK: Loading
[LNKXCT LAST execution]

2	1	10
1	2	-1
1	4	1
1	2	2
1	4	-2

STOP

END OF EXECUTION

CPU TIME: 0.04 ELAPSED TIME: 0.44

EXIT

.KJOB/BATCH

[LGRTAJL Another job is still logged-in under [15100,150064]]
 Job: 17 User: MANOJ KUMAR [15100,150064]
 Logged-off PTY115 at 3:23:31 on 29-Nov-85
 Runtime: 0:00:00, KCS:10, Connect time: 0:00:11
 Disk Reads:234, Writes:3
 SARCON version 102(2067) running LASR sequence 2447 in stream 1
 Input from DSKC:LAST.CTL[15100,150064]
 Output to DSKC:LAST.LOG[15100,150064]
 Job parameters Core:100P Unique:YES Restart:YES Output:VOLUG
 Time:00:01:00

LOGIN 15100/150064 /DEFER/SPJOB:AUO/TIMES:60/CBRE:100P/LOCATE:10/NAME:"MANDJ KUMAR"
JDB 13 I I F KANPUR 603A(3) TTY115
[LGN JSP Other jobs same PPN:24]
0325 29-Nov-86 Sat

TY FOR 22 DAT

[3:25:13]

12345
05444
1100000000000000

EX-LAST-FDR

[3:25:13]

LINK: Loading
[LNKXCT LAST execution]

123456789
000001212
00000424242
444111111
22222222222

25

END OF EXECUTION

CPU TIME: 0.06 ELAPSED TIME: 0.36

EXIT

•KJQB/BATCH

[DGRAJL] Another job is still logged-in under [15100,150064]
Job 13 User MANOJ KUMAR [15100,150064]
Logged-off RTY115 at 3:25:16 on 29-Nov-85
Runtime: 0:00:00, KCS:7, Connect time: 0:00:07
Disk Reads:181, Writes:3

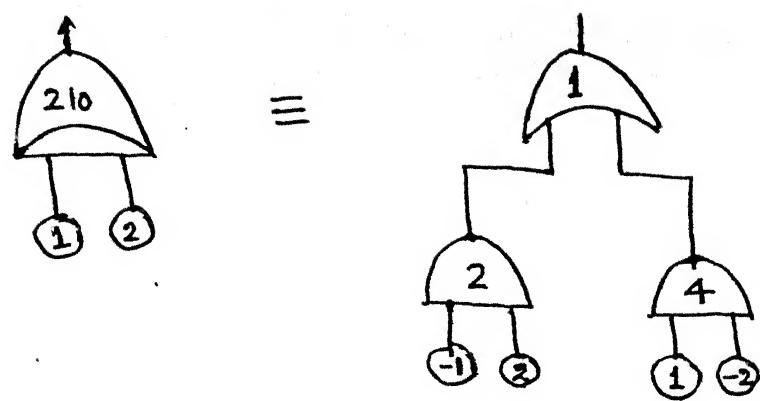


Figure : Example 1

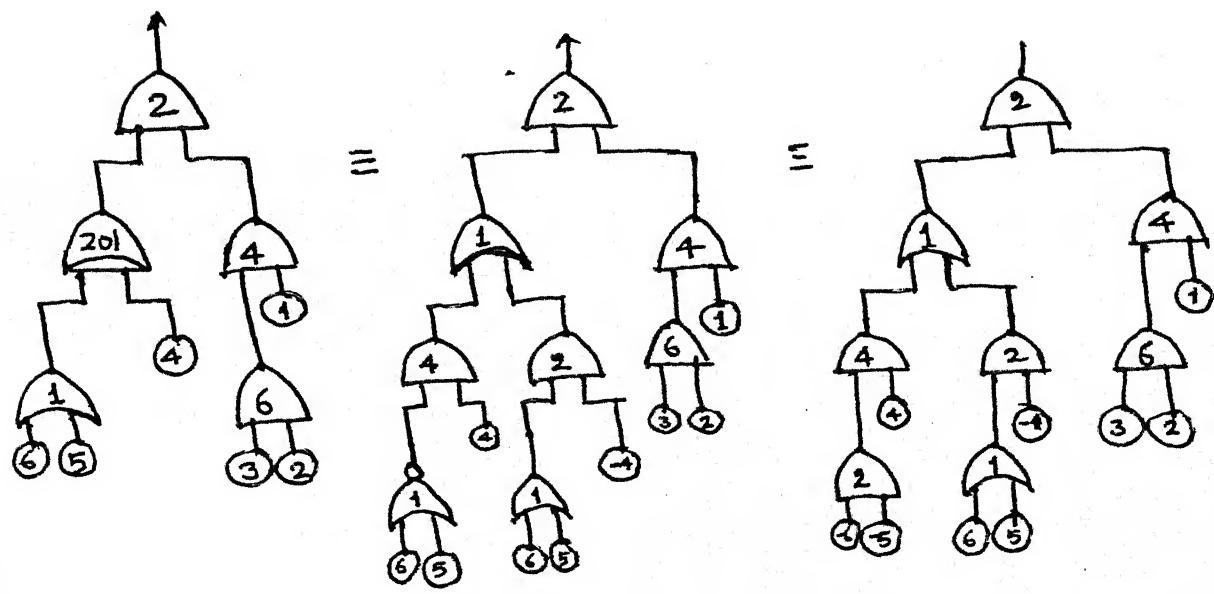


Figure : Example 2

For MODTREE

A 98977

TH
005.3 Date Slip A 98977

K964 This book is to be returned on the date last stamped.

R96 This book is to be returned on the

NETP-1987-M-KUM-AUT